Optimization of the workflow in a BOINC-based Desktop Grid for virtual drug screening

Natalia Nikitina[1] [0000−0002−0538−2939] and
Evgeny Ivashko[1,2] [0000−0001−9194−3976]

[1] Institute of Applied Mathematical Research, Karelian Research Center of the Russian Academy of Sciences, Pushkinskaya 11, 185910, Petrozavodsk, Russia
[2] Petrozavodsk State University, Lenina 33, 185035, Petrozavodsk, Russia
nikitina@krc.karelia.ru, ivashko@krc.karelia.ru

Abstract. This paper presents an analysis of a BOINC-based volunteer computing project SiDock@home. The project implements virtual drug screening. We analyse the employed workflow describing the processes of task generation, results creation, validation and assimilation. Basing on this analysis, we propose an optimized workflow aimed at minimization of computing intensity and scaling up the granularity of the results.

Keywords: Distributed computing · Volunteer computing · Desktop Grid · Task scheduling · BOINC · Virtual screening · Molecular docking.

1 Introduction

Desktop Grid is a high-throughput computing paradigm based on using the idle time of non-dedicated geographically distributed general-purpose computing nodes (usually, personal computers) connected to the central server by the Internet or a local access network. The concept of Desktop Grid was introduced in 1987 [21] and used in many projects since then. Today, such a computing paradigm allows to efficiently solve computationally intensive scientific problems in mathematics (Gerasim@home [9], Amicable Numbers [2]), biology (Rosetta@home [30], Folding@home [34]), physics (LHC@home [17]), astronomy (Einstein@Home [7], Universe@Home [35]) and other areas of science.

Among the variety of high-performance and high-throughput computing systems, Desktop Grids hold a special place due to their enormous potential and, at the same time, high availability. For instance, in 2020, a volunteer computing project Folding@home gathered resources exceeding one exaflops and became the first world's exascale system, more powerful than the top 100 supercomputers combined [36]. The overall potential of Desktop Grids is estimated as hundreds of exaflops [3], exceeding the total power of all existing supercomputers. It makes Desktop Grid a credible alternative to other high-performance and high-throughput computing systems when solving urgent scientific problems [1].

In more than 30 years of the existence of Desktop Grids, multiple software platforms have been implemented and used for their operation. The high availability and ease-of-use of Desktop Grids complicate the enumeration and analysis of the variety of Desktop Grid-based projects. However, recent reviews of the public volunteer computing projects [20,6] show that up to 80% of them are based on BOINC (Berkeley Open Infrastructure for Network Computing) [3].

BOINC is an Open Source software platform for Desktop Grid deployment with a client-server architecture (Fig. 1). The server consists of a number of parallel services sharing a database (see [3] for detailed architecture). The client connects to the server to request tasks, performs computations and sends the results back to the server. The server validates and assimilates the results, aggregating them to obtain the solution of the initial scientific problem.

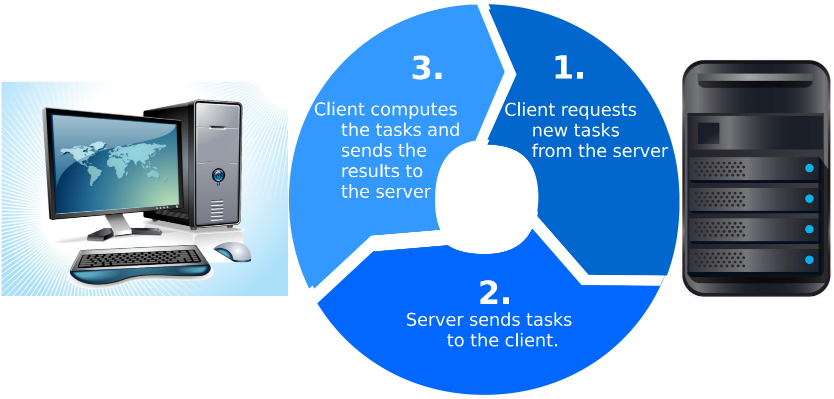
3.
Client computes the tasks and sends the results to the server
1.
Client requests new tasks from the server
2.
Server sends tasks to the client.

Fig. 1: Client-server architecture of BOINC.

In high-performance and high-throughput computing, the term scientific workflow generally stands for a data-intensive and computationally intensive scenario with interdependencies between parallel tasks [18]. Such a scenario is typically modeled in the form of a directed acyclic graph representing the tasks and dependencies between them. Optimization of scientific workflows is a broad subject that has been rigorously studied in the literature [5,18,10,15]. In BOINC-based Desktop Grids, however, the tasks are usually mutually independent and follow the model of “bag-of-tasks”. In the present work, we restrict ourselves to such a model of a workflow. Concrete scientific problems and computational platforms impose specific requirements on the workflow optimisation. This is the case, in particular, for BOINC-based biomedical research [16,11,29].

Desktop Grid are aimed at gathering large amount of resources for a long period of time, so as to repeatedly process huge numbers of homogeneous tasks. With that, even small improvements could save years of CPU time. In this paper, we discuss the problem of performance optimization of a Desktop Grid-based computing project performing large-scale molecular docking. We perform analysis of the computational process, show that the issues of redundant computations and correctness are of high importance for such a project, and propose a design of the workflow which will address these issues.

The paper is organized as follows. In Section 2, we describe the scope of volunteer computing project SiDock@home aimed at drug discovery via high-throughput virtual screening. In Section 3, we summarise and analyse the computational efficiency of the project and propose a solution to the identified problems. In Section 4, we comment on the immediate implementation of the proposed solution. Finally, in Section 5, we conclude the paper.

2 Volunteer computing project SiDock@home

2.1 Project setup

In March 2020, a citizen science project "*Citizen science and the fight against the coronavirus*" (COVID.SI) [12] was initiated in the field of drug design and medicinal chemistry. The project is aimed at drug discovery, first of all, against coronavirus infection, using high-throughput virtual screening (HTVS) [19,33] on a small molecules library developed by the team. The problem of HTVS has the bag-of-tasks type and can be efficiently implemented in a Desktop Grid environment. At the same time, the scale of the problem requires a significant amount of computational resources that increases proportionally to the size of the small molecules library. These factors made volunteer computing an efficient tool for COVID.SI. The project received a good response from the community and showed the applicability of the volunteer computing to the ongoing research.

In the following months, SiDock@home, a BOINC-based extension of COVID.SI, was created and grew into a sizable, independent and competent research project for general drug design. We describe the project setup in detail in [25,27] and estimate its performance in [26]. As of the end of March, 2022, there are 7540 registered users in SiDock@home providing about 8000 active computers with the total processing power 56 Teraflops as measured by means of BOINC.

2.2 High-throughput virtual screening

Drug development is a time-consuming and expensive process which takes up to 12-17 years. In general case, it starts from the identification and validation of a biological *target*, a molecule which is associated with the disease and whose activity is expected to be modulated by a drug for a desired therapeutic effect.

The second stage is identification of *hits*, chemical compounds with high predicted binding activity against the target. The set of hits is subject to analysis and optimization to obtain *leads*, compounds that have several desired attributes in addition to high binding activity. Next stages of drug development are the additional optimization of leads, preclinical and clinical studies, registration of the new drug and postmarketing studies.

Identification of hits is a specific resource-consuming problem which may be seen as a funnel where a large library of chemical compounds is reduced to a set of $10-10^3$ hits to be tested in a laboratory [33]. HTVS allows to identify hits *in silico* so as to reduce time and cost of the first stages of drug development.

For this purpose, hits are selected basing on molecular docking of the library of chemical compounds against the target.

In addition, identification of hits is a complex problem. There are many possible criteria for the selection of hits, and hardly any of them are absolute. The computational process should allow a flexible management of HTVS with the possibly minimal time to obtain a set of hits with desired characteristics.

3 Analysis and optimization of the computational process

Performance is a permanent concern of a high-performance computing project. We believe that every developer should try to get the maximum out of available resources. At the same time, the workflow must be flexible enough to support the scalability of the computational project, which is especially important in case of volunteer computing resources known for their irregular nature (Fig. 2, 3).

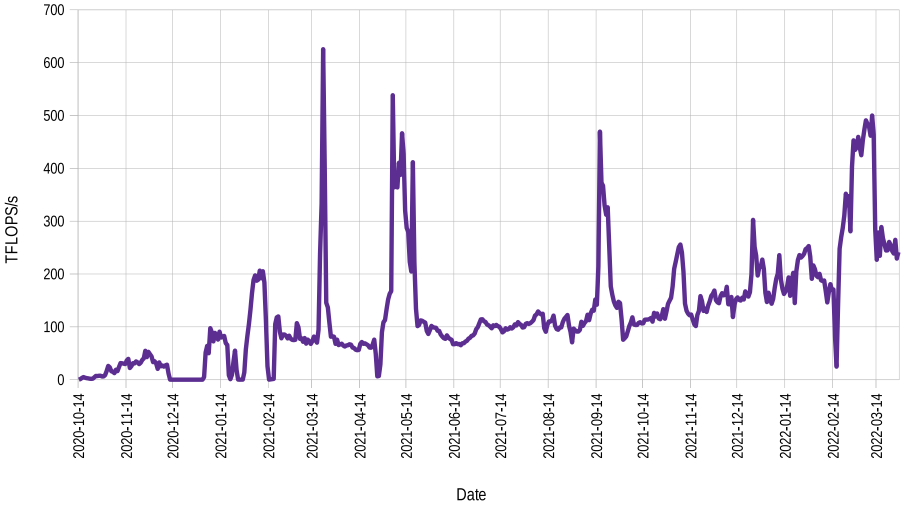

Fig. 2: Performance of SiDock@home, FLOPS/s, measured by means of BOINC.

Having in background a 1.5-year experience of running SiDock@home project, we analyze its computational process, aiming to obtain the set of hits faster and by spending less resources. The analysis is presented below.

3.1 Analysis of a conventional computational process

The workflow of HTVS consists of mutually independent tasks, each of which performs molecular docking of one or more small molecules to a specified target. The process of molecular docking is illustrated in Fig. 4 prepared with PyMOL

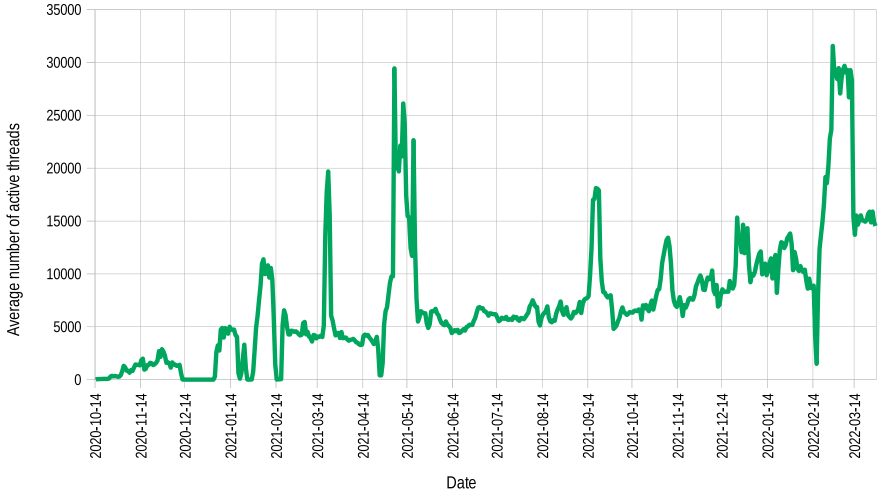

Fig. 3: Performance of SiDock@home, threads, measured by means of BOINC.

software [32]. As a result of molecular docking, the small molecules are ranked according to the predicted binding affinity.

There exist a number of software products for molecular docking with various algorithms. In SiDock@home, we employ molecular docking software CmDock [4,14] which started in 2020 as a fork of an open-source software RxDock [31,23], aimed at optimisation, implementation of new features and utilisation of modern hardware.

The heterogeneous and unreliable nature of Desktop Grid implies that a result of the task may be erroneous or never be returned to the server. In order to obtain correct results in a fixed period of time, the BOINC platform implements several mechanisms that are described in detail in [3,13]. In short, they are:

- deadline: time moment after which a task is considered lost;
- replication: issuing of several instances of the same task;
- quorum: the number of matching answers to consider a result valid.

In this section, we focus our attention on the tasks that have been executed at the client side correctly and within deadline. The results of such tasks provide the progress on the scientific problem being solved in the Desktop Grid.

In SiDock@home, the life cycle of such a task is, for the most part, the same as in other BOINC-based projects performing HTVS (for instance, the OpenZika project within World Community Grid [8]).

Task generation. The task is generated at the server side and includes an input file with a description of the target, docking parameters and a list of small molecules. If there is need for replication, the same task is created in two or

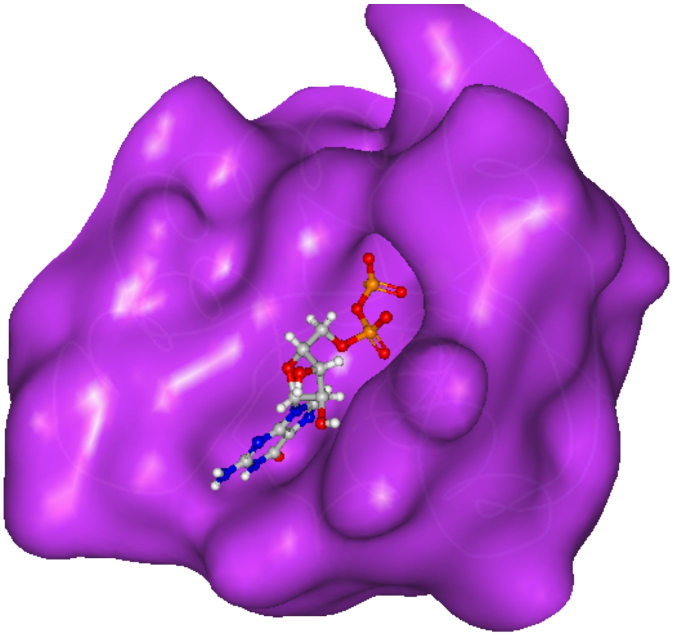

Fig. 4: Molecular docking of a small molecule to a target (in purple).

more instances, and the value of quorum Q is specified. The generated task is put into the queue. When a client requests work from the server, it receives a portion of tasks from the queue.

Task execution. At the client side, the following steps are taken for each molecule in a task:

1. A special software program performs molecular docking against the target in a limited number of runs. At each run, the program finds a docking pose and calculates the score based on the predicted binding energy.
2. If the best found score exceeds the given threshold, it is written into the output file together with the corresponding pose.

The result is a list of docking poses of the compounds with the score exceeding the threshold. Such a pre-filtration allows to omit the results with low predicted binding affinity so as to reduce the amount of transferred data which is important for Desktop Grid systems. In addition, a log file keeps the key information about all docked molecules in order to verify the result integrity.

The client sends ready results to the server upon the next communication.

Result validation. In order to process the result, the server needs to verify that it comes from a task which has been correctly executed at the client side.

For this purpose, BOINC validator checks the presence and format of output files. A result is considered valid if it has a positive integer number of the docked molecules logged.

If replication is used, the next step is to check if the results of two instances of a task are equivalent. Two valid results are considered equivalent if they have coinciding lists of the docked molecules logged. This means that all the input molecules were processed during the docking. If quorum Q has been reached, one of the valid results is considered canonical and is stored for further processing. Each valid result is awarded an equal amount of BOINC credits depending on task runtime.

Result assimilation. Output files of canonical results are placed to a special directory at the server for further processing.

When HTVS has been performed over the whole library of small molecules against a specified target and the results have been assimilated, they undergo partially automated processing and expert analysis.

Currently, there are 500 molecules in a task in SiDock@home. The whole library contains description of 982 401 500 molecules, so it is covered by 1 964 803 tasks. Such a division of the library allows us to have an average task runtime of about 1-2 hours on a typical desktop computer, following the common practice of volunteer computing. Quorum $Q = 2$ is used. With such settings, the library of small molecules has been screened against 11 potential therapeutic targets during the first 17 months of the project operation. The quorum allowed to recognize a number of technical errors and problematic entries of small molecules in a semi-automated way and to update the docking application, the library of small molecules and server-side programs accordingly.

3.2 Known drawbacks of the conventional computational process

Due to replication and unreliability of the Desktop Grid nodes, there is an overhead in terms of tasks needed to obtain valid results. In Table 1, we present this overhead for five targets. At the time of the experiments, the screening of the whole library comprised 1 966 935 tasks for each target. On top of that, a number of tasks were added:

- *Excessive tasks.* These tasks ended with a computational error, validation error or were discarded because another result was accepted as canonical.
- *Lost tasks.* Results of these tasks were never received from a client, most likely because it left the project.

Such tasks have consumed computational resources but did not directly contribute to the hit identification.

Due to the randomized character of computations, the results of two replicas of the same task may not coincide regardless of technical errors or cheating. This is often the case in molecular simulations [16].

Target	Excessive tasks	Lost tasks
PLpro_v2	116.29%	0.35%
RdRp_v1	104.55%	0.71%
3CLpro_v4	100.87%	0.22%
3CLpro_v5	101.22%	0.32%
3CLpro_v6	100.87%	0.52%

Table 1: Overhead in terms of tasks. The basic number is 1 966 935 tasks (100%).

Recall that a task contains 500 molecules to dock. For a variety of reasons, the docking of some molecules in a task can fail. The result of such a task will be considered invalid, which leads to a wasting of time and computing power. Earlier, we considered a related problem in [22]. When small molecules are packed into "parcels" to form a task, too small a size of the parcel means a very high number of communications between the client and the server and, consequently, a high load on the server. Too large a size of the parcel, on the other hand, means that a high amount of computational time will be wasted in case of a single error. To calculate the optimal size of the parcel, we proposed a game-theoretical mathematical model. For an enterprise-level Desktop Grid, the derived solutions were shown to decrease server load by a factor of 3.

In the present work, we consider several factors in addition to the server load, and elaborate on a more reliable approach which is described in subsection 3.3 and will allow to save the intermediate docking results of separate molecules.

In the new approach, the task size remains an important issue. The size of 500 has been selected empirically to obtain a moderate average runtime. But the performance of individual Desktop Grid clients may vary a lot, and different targets may have different runtimes. Additionally, there are periods of an increased server load, such as the competitions in the BOINC community. A dynamically adjusted task size would allow to balance the load through generation of longer-running tasks when needed.

Another important question is the communication between the client and the server. Each client generates a number of requests to the server to obtain new tasks and send back the results. The output file may have a significantly larger size than the input data. The server load caused by these communications may be decreased, firstly, by an increase in the task size (as mentioned above), and secondly, by the transfer of output files to a dedicated file server.

Finally, the organization of the workflow influences the scope of results subject to expert evaluation, namely the set of identified hits. A docking configuration of a hit is important, but it is not reasonable to store all the docking results because it would require the storage and transfer of large amounts of data (about 0.7 terabytes per target), most of which will not pass a primary threshold. For this reason, there is a preliminary filtration of results at the client's side. It allows to reduce the necessary amount of data, but comes at the price of losing information about molecules with low scores. If the threshold has been set too high, there may be a need to rescreen the whole library, leading to redundant use of computational resources and delaying testing of the results in a laboratory.

Let us summarize the problems described above:

- overhead due to the quorum $Q = 2$;
- wasting of time and computing power in case of a single fail in docking;
- a fixed size of a task;
- amount of client-server communications,
- scope of results.

In the next subsection we propose the solutions to the listed problems.

3.3 Optimization of the computational process

The current approach used in SiDock@home allows to automate, for the most part, the validation of results and identification of technical errors. At the same time, it has disadvantages that lead to a large amount of duplicated computations and, overall, slow down the hit discovery rate and limit the scope of hits selection. In this section, we describe a new organization of the computational process.

In the new approach, the workflow will base on a summary table describing the HTVS process and consisting of entries

```
<target_id, ligand_id, score, task_id>.
```

Here, `target_id` identifies the target of the virtual screening, `ligand_id` identifies a small molecule docked against the target, `score` contains the calculated docking score, and `task_id` identifies the computational task within which the docking score was calculated. The workflow will change as follows.

Task generation. New tasks will be generated dynamically so as to include the molecules that have not been docked yet and form tasks of a desired length. Initially, only one instance of each task will be generated, and the quorum will be set to $Q = 1$. A list of molecules belonging to the task will be recorded at the server side.

Task execution. At the client side, the resulting score will be logged for every small molecule. For the molecules with the score exceeding the threshold, the best pose will be written into the output file.

Result validation. Instead of comparing the replicas, a result will be considered valid if it contains a positive number of successfully docked ligands from the recorded list of molecules associated with this task.

Result assimilation. Target ID, ligand IDs and the calculated scores will be written into the summary table. Output files of validated results will be placed to a special directory at the server for further processing.

Upon completion of the virtual screening for a specified target, the summary table will contain consolidated information on the docking results for each target. It will support the expert selection of the set of hits with desired characteristics such as the number, chemical diversity, lead-likeness etc. Different hit cutoff metrics can be used for the selection, and some of them may be calculated basing on the present data without the need to repeat molecular docking. An example is ligand efficiency: the predicted binding energy normalized by the number of heavy atoms in the molecule.

Correct results will be accepted at the quorum of one, which will eliminate the need for duplication of computations and speed up the HTVS process. The size of new tasks can be adjusted for computational experiments of different runtimes. New tasks can be easily generated to dock selected molecules with complementary methods. For example, computationally expensive simulations may be performed for the top selected molecules.

In this way, the summary table will allow to better design new HTVS experiments. For example, one can create a representative sample of the library to run an initial screening and adjust the next HTVS process accordingly. With dynamic task generation, HTVS may be easily designed so as to prioritize a set of the most prospective molecules [28], balance between the number and diversity of hits [24], etc.

4 Implementation and results

With the new approach, the architecture of the project database will be modified so as to include the summary table and auxiliary tables. The program code of BOINC validator and assimilator will be altered so as to implement the proposed approach. Other processes on the server side will remain unchanged.

As an intermediate step, the proposed design of the workflow can be implemented together with the existing mechanism of adaptive replication in BOINC. With adaptive replication, the results obtained by reliable hosts are accepted at the quorum $Q = 1$. If a result comes from an unreliable host, another replica of the task is issued and the quorum is set to $Q = 2$. Reliability is based on the number of consecutive valid results returned by the host and is constantly updated. The concept of validity depends on the considered scientific problem and the implementation of the computational process. The implementation described in subsection 3.3 allows to set any quorum without loss of reliability.

At the moment, the proposed design of the workflow has been implemented in an accompanying BOINC project operating for testing purposes. The clients in this project are heterogeneous computers belonging to the team of authors. The new approach is being tested and adjusted.

5 Conclusion

Virtual screening assists drug discovery and allows to reduce time and cost of its earliest stages. At the same time, it is a resource-demanding problem which usually requires a high-throughput computational infrastructure. An appropriate design of the workflow should allow efficient use of such an infrastructure.

Peculiar features of Desktop Grids make it necessary to control the reliability of results while ensuring their fastest discovery and rational use of the participating computers. Design and implementation of the workflow depends on the specific scientific problem being solved in the Desktop Grid.

In this paper, we analyze the workflow in a volunteer computing project SiDock@home and propose its optimization. We hope that the proposed design of the workflow will be of interest to other researchers at their computational experiments on virtual screening. In future work, we plan to investigate the performance of SiDock@home with the new workflow and derive new mathematical models and algorithms of task scheduling that will contribute to the further development of Desktop Grids.

References

1. Alnasir, J.J.: Distributed Computing in a Pandemic: A Review of Technologies Available for Tackling COVID-19. arXiv preprint arXiv:2010.04700 (2020)
2. Amicable Numbers. https://sech.me/boinc/Amicable/, [Online; accessed 12-Apr-2022]
3. Anderson, D.P.: BOINC: a platform for volunteer computing. Journal of Grid Computing **18**, 99–122 (2020). https://doi.org/10.1007/s10723-019-09497-9
4. Bahun, M., Jukić, M., Oblak, D., Kranjc, L., Bajc, G., Butala, M., Bozovičar, K., Bratkovič, T., Podlipnik, Č., Poklar Ulrih, N.: Inhibition of the SARS-CoV-2 3CLpro main protease by plant polyphenols. Food Chemistry **373**, 131594 (2022). https://doi.org/https://doi.org/10.1016/j.foodchem.2021.131594, https://www.sciencedirect.com/science/article/pii/S0308814621026005
5. Deelman, E., Peterka, T., Altintas, I., Carothers, C.D., van Dam, K.K., Moreland, K., Parashar, M., Ramakrishnan, L., Taufer, M., Vetter, J.: The future of scientific workflows. The International Journal of High Performance Computing Applications **32**(1), 159–175 (2018)
6. Distributed Computing – Computing Platforms. http://distributedcomputing.info/platforms.html, [Online; accessed 12-Apr-2022]
7. Einstein@Home. https://einsteinathome.org, [Online; accessed 12-Apr-2022]
8. Ekins, S., Perryman, A.L., Horta Andrade, C.: OpenZika: An IBM World Community Grid Project to Accelerate Zika Virus Drug Discovery. PLOS Neglected Tropical Diseases **10**(10), 1–5 (10 2016). https://doi.org/10.1371/journal.pntd.0005023, https://doi.org/10.1371/journal.pntd.0005023
9. Gerasim@home main page. https://gerasim.boinc.ru/, [Online; accessed 12-Apr-2022]
10. Ghafarian, T., Javadi, B., Buyya, R.: Decentralised workflow scheduling in volunteer computing systems. International Journal of Parallel, Emergent and Distributed Systems **30**(5), 343–365 (2015)

11. Ghorbani, M., Swift, S., Taylor, S.J., Payne, A.M.: Design of a flexible, user friendly feature matrix generation system and its application on biomedical datasets. Journal of Grid Computing **18**(3), 507–527 (2020)
12. Home – COVID.SI. `https://covid.si/en`, [Online; accessed 12-Apr-2022]
13. Ivashko, E., Chernov, I., Nikitina, N.: A survey of Desktop Grid scheduling. IEEE Transactions on Parallel and Distributed Systems **29**(12), 2882–2895 (2018)
14. Jukič, M., Škrlj, B., Tomšič, G., Pleško, S., Podlipnik, Č., Bren, U.: Prioritisation of Compounds for 3CLpro Inhibitor Development on SARS-CoV-2 Variants. Molecules **26**(10) (2021), `https://www.mdpi.com/1420-3049/26/10/3003`
15. Juve, G., Chervenak, A., Deelman, E., Bharathi, S., Mehta, G., Vahi, K.: Characterizing and profiling scientific workflows. Future generation computer systems **29**(3), 682–692 (2013)
16. Leguy, J., Glavatskikh, M., Cauchy, T., Da Mota, B.: Scalable estimator of the diversity for de novo molecular generation resulting in a more robust QM dataset (OD9) and a more efficient molecular optimization. Journal of cheminformatics **13**(1), 1–17 (2021)
17. LHC@home. `https://lhcathome.web.cern.ch`, [Online; accessed 12-Apr-2022]
18. Liew, C.S., Atkinson, M.P., Galea, M., Ang, T.F., Martin, P., Hemert, J.I.V.: Scientific workflows: moving across paradigms. ACM Computing Surveys (CSUR) **49**(4), 1–39 (2016)
19. Lionta, E., Spyrou, G., Vassilatis, K.D., Cournia, Z.: Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances. Current Topics in Medicinal Chemistry **14**(16), 1923–1938 (2014). https://doi.org/10.2174/1568026614666140929124445, `http://www.eurekaselect.com/article/62572`
20. List of distributed computing projects. `https://en.wikipedia.org/wiki/List_of_distributed_computing_projects`, [Online; accessed 12-Apr-2022]
21. Litzkow, Michael J.: Remote Unix: Turning idle workstations into cycle servers. In: Proceedings of the Summer USENIX Conference. pp. 381–384 (1987)
22. Mazalov, V.V., Nikitina, N.N., Ivashko, E.E.: Hierarchical two-level game model for tasks scheduling in a Desktop Grid. In: 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). pp. 541–545 (2014). https://doi.org/10.1109/ICUMT.2014.7002159
23. Morley, S.D., Afshar, M.: Validation of an empirical RNA-ligand scoring function for fast flexible docking using RiboDock®. Journal of computer-aided molecular design **18**(3), 189–208 (2004)
24. Nikitina, N., Ivashko, E., Tchernykh, A.: Congestion game scheduling for virtual drug screening optimization. Journal of computer-aided molecular design **32**(2), 363–374 (2018)
25. Nikitina, N., Manzyuk, M., Jukić, M., Podlipnik, Č., Kurochkin, I., Albertian, A.: Toward Crowdsourced Drug Discovery: Start-Up of the Volunteer Computing Project SiDock@home. In: Russian Supercomputing Days. pp. 513–524. Springer (2021)
26. Nikitina, N., Manzyuk, M., Podlipnik, Č., Jukić, M.: Performance estimation of a BOINC-based Desktop Grid for large-scale molecular docking. In: International Conference on Parallel Computing Technologies. pp. 348–356. Springer (2021)
27. Nikitina, N., Manzyuk, M., Podlipnik, Č., Jukić, M.: Volunteer Computing Project SiDock@home for Virtual Drug Screening Against SARS-CoV-2. In: International Conference on Computer Science Protecting Human Society Against Epidemics. pp. 23–34. Springer (2021)

28. Pradeep, P., Struble, C., Neumann, T., Sem, D.S., Merrill, S.J.: A novel scoring based distributed protein docking application to improve enrichment. IEEE/ACM transactions on computational biology and bioinformatics **12**(6), 1464–1469 (2015)
29. Quang, B.T., Kim, J.S., Rho, S., Kim, S., Kim, S., Hwang, S., Medernach, E., Breton, V.: A comparative analysis of scheduling mechanisms for virtual screening workflow in a shared resource environment. In: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp. 853–862. IEEE (2015)
30. Rosetta@home. `https://boinc.bakerlab.org`, [Online; accessed 12-Apr-2022]
31. Ruiz-Carmona, S., Alvarez-Garcia, D., Foloppe, N., Garmendia-Doval, A.B., Juhos, S., Schmidtke, P., Barril, X., Hubbard, R.E., Morley, S.D.: rDock: A fast, versatile and open source program for docking ligands to proteins and nucleic acids. PLoS computational biology **10**(4), e1003571 (2014)
32. Schrödinger, LLC and Warren DeLano: PyMOL. `http://www.pymol.org/pymol`, version 2.4.0, 2020-05-20
33. Tanrikulu, Y., Krüger, B., Proschak, E.: The holistic integration of virtual screening in drug discovery. Drug Discovery Today **18**(7-8), 358–364 (2013)
34. Together We Are Powerful – Folding@home. `https://foldingathome.org`, [Online; accessed 12-Apr-2022]
35. Universe@Home. `https://universeathome.pl/universe`, [Online; accessed 12-Apr-2022]
36. Zimmerman, M.I., Porter, J.R., Ward, M.D., Singh, S., Vithani, N., Meller, A., Mallimadugula, U.L., Kuhn, C.E., Borowsky, J.H., Wiewiora, R.P., et al.: SARS-CoV-2 simulations go exascale to predict dramatic spike opening and cryptic pockets across the proteome. Nature chemistry **13**(7), 651–659 (2021)