A Web-based System for Launching Large Experiment Series on Supercomputers*

Evgeniy Kuklin[1,2] and Sergei Pravdin[1,2]

[1] Krasovskii Institute of Mathematics and Mechanics, Ekaterinburg, Russia
key@imm.uran.ru
[2] Ural Federal University, Ekaterinburg, Russia

Abstract. The researchers developed a user-friendly web-based service for launching large series of experiments on parallel computing systems. Simulation of various biological processes requires that dozens of numerical experiments with parameter variations have to be conducted. A key feature of the proposed system is the automatic generation of configuration files based on a proposed algorithm for generating tuples and automatic job launching, which saves researchers from manual data preparation. The developed system is currently used for conducting computational experiments to study the drift of spiral waves in myocardium on the *Uran* supercomputer, Ekaterinburg, Russia.

Keywords: Parallel computing systems · Supercomputers · Experiment series launching · Graphical user interface · Living system simulation

1 Introduction

Living system simulation often demands a large number of computational experiments on the same model but with varying parameter values. Since experiments are time-consuming, conducting them in a reasonable time frame without parallel computing systems and supercomputers is difficult. In order to obtain accurate results, dozens or hundreds of numerical experiments must be prepared and performed. For many scientists, working with a command line on supercomputers is tedious and frustrating.

Often, these problems are solved using already existing specialized applications, which are accessed from various web platforms. However, most platforms are focused on a specific task and utilize preinstalled software, which limits their use. In addition, they do not allow users to flexibly configure the automatic generation of input data for different simulation models. This can cause problems when conducting a large number of experiments.

To reduce experiment preparation, the authors developed the web-based multiplatform system for launching experiments on a supercomputer. The system conducts large series of computational experiments with different parameter values, saving scientists from manually preparing the input data and launching jobs.

* Supported by the RSF project 17-71-20024 (IMM UB RAS).

It tracks and saves parameters of previous experiments to be reused or reconfigured. The service provides a simple web interface that works remotely without the installation of additional software. Though the project was initiated for the purposes of heart modeling, the system can also be used for different computational clusters and can run almost any software that has an input configuration file in the standard INI file format. A build-in parser supports sections, enclosed in square brackets and making parameter lists more structured, and comments. If varied parameter values make an arithmetic progression, they can be specified in a short form as a start value, an increment, and a final value. The present article describes the architecture of the developed system, as well as the algorithm for generating tuples from sets of values of individual parameters, and features of the graphical user interface.

2 Service Architecture

After we encountered technical problems when using Python for desktop GUI development, we conceptualized the interface. Remote users could use different operation systems, and could not have administrator rights, which caused problems with installing the required packages for the desktop version. Therefore, a web-based interface was conceived, which should work with any computer, even on a tablet or smartphone.

The schematic diagram of the system is shown in Fig. 1. The main components of the system are located on a dedicated server. These include: a web server, which provides the interface, a database, a storage directory for experimental metadata, and several service scripts.

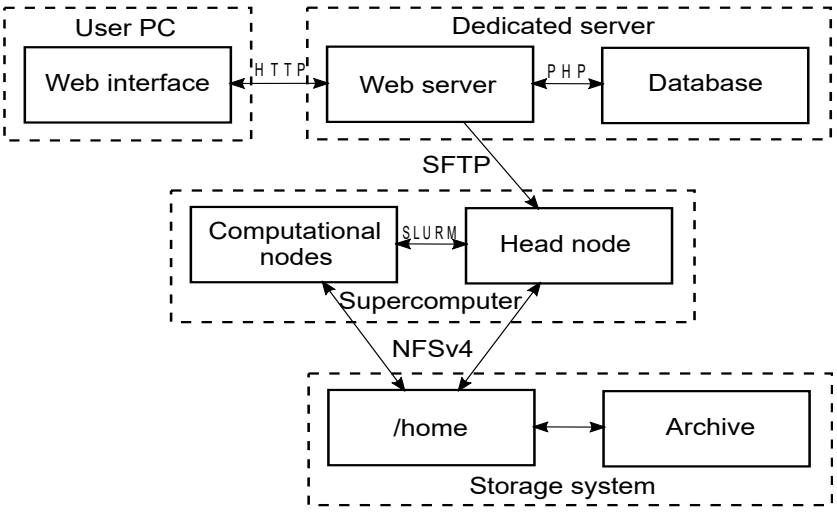


Fig. 1. Web service architecture

The system uses the concept of a computational package, which is prepared via the interface and then sent through "SFTP" protocol to the supercomputer storage system into the user's home directory. The entire computational package is stored in a special directory on the web server for possible re-use in the future; key information about the package is written to the database. All supporting scripts are sent along with the original data in the computational package, but they can also be taken from a special directory on the storage system.

On the supercomputer, after the automatic preparation of input configuration files for each experiment of the series (see Section 3.3), the launch script sets the tasks for execution through interacting with the supercomputer workload manager, such as "SLURM". Data obtained as a result of the calculations remain in the storage system, which is connected to the compute nodes and the cluster head node via "NFSv4" protocol. These data can optionally be placed in a common archive of the research group, which work with the system.

3 Technical Issues

3.1 Web Interface

One of the main objectives of the project was to provide an easy-to-use system through a user-friendly interface requiring minimal adaptation. As mentioned, the universal web interface used likely on any computer.

The process of setting a series of experiments is divided into four steps. In the first step (see Appendix; Fig. 2), a user can start a new project, select one of the previous projects, or check the status of jobs already running on the supercomputer.

The second step (Appendix; Fig. 3) involves editing the configuration file for the simulation program. Simulation of living systems may involve the launch of a large number of experiments on one model with a variation of several parameters. For convenience, the parameter values for each experiment can be specified in one file, separating the values with a space.

In the third step (Appendix; Fig. 4), the user specifies all the necessary information to place the jobs in the execution queue on a parallel computing system. The information includes the path to the source folder, parameters for the workload manager, the working directory, and a description of the experiment. Presumable, the source code is already stored on the supercomputer, since the program must be built there.

The last step (Appendix; Fig. 5) displays details of the experiment series, and the queue of the active user jobs (confirming the tasks have been successfully started for calculation). To display the queue, the supercomputer workload manager receives a request with the appropriate command from the web server.

3.2 Database Usage

In order to repeat a previous experiment, it is sufficient to save the simulation program, the experimental configuration files, and the parameters for the workload manager. Every experiment is assigned a unique identifier stored in the

database along with other parameters of the experiment that becomes the name of the directory storing the files related to the experiment. The rest of the data is saved in the next steps of the experiment.

On the service home page, the latest experiments of all users are displayed using information from the database (see Appendix; Fig. 2). This makes it possible to avoid duplicate calculations when required experiment has already been conducted or to reproduce an experiments when necessary. Moreover, once performed, a series of experiments can be easily restarted with different parameters without replacing the fields, which accelerates launching repeated calculations.

The database used for the project is PostgreSQL [1]. The launching scripts were written in Python, Bash, and PHP.

3.3 Generation of Configuration Files

The used simulation software stores all model parameters in a standard INI configuration file. For example, to study the drift of spiral waves in the myocardium, several models with 10–20 parameters are used, and almost every parameter in the corresponding configuration files can be variated. The system never knows how many combinations will result from user settings. Using a recursion is not a good idea, so an efficient algorithm for full parameter enumeration was necessary.

Each input file describes values of parameters $a_0, a_1, \ldots, a_k$. According to a specific task, a parameter a_i can be constant for all launches or must take values from a set $V_i = \{a_i^0, a_i^1, \ldots, a_i^{n_i}\}$. So, a set of input files where varying parameters take values from their respective sets must be created. We propose a simple algorithm to get all tuples. The number of tuples is $N = (n_0 + 1)(n_1 + 1) \ldots (n_k + 1)$. Each input file is created based on indices $i_0, i_1, \ldots, i_k$ of parameter values. To get a tuple corresponding to an index $I \in \{0, \ldots, N-1\}$, we use a backing array $J_{1\ldots k}$ and utilize the algorithm 1.

Algorithm 1: Tuples generation algorithm

procedure FINDTUPLES
$J_k = I$
for p from $k-1$ downto 1 do
begin
$i_p = J_{p+1} \bmod n_p$
$J_p = J_{p+1} \operatorname{div} n_p$
end
$i_0 = J_1 \bmod n_0$

Finally, the I^{th} tuple is formed using indices $i_0, i_1, \ldots, i_k$, which serve to create N configuration files with unique parameter values. Each configuration file is placed to its own directory, from where its own copy of simulation software is launched. For convenience, the directory names depend on values and names of the varied parameters.

Two ways of specifying a set of values of one parameter are supported. Users may simply list the values by separating them with a space or apply an expression, like 200...10...300, in which the initial value, the step, and the final value are specified, respectively.

4 Discussion and Approbation

The users of the developed system appreciated the convenience of the new web GUI and the ability to launch a series of dozens of experiments in five minutes without working in a command line. Using settings from previous experiments reduces the time to launch a new experiment series to less than a minute. Overall, the user-friendly system helped the researchers to conduct computational experiments more efficiently.

The proposed system was tested with the *Uran* supercomputer located at the Krasovskii Institute of Mathematics and Mechanics of the Russian Academy of Sciences in Ekaterinburg, Russia. The system currently carries out computational experiments to study the drift of spiral waves in myocardium. Myocardium is an active medium and consists of interconnected elements. They have a resting state and can temporarily reach an excited state when enough stimulus is applied. Excited elements produce stimuli that spread in all directions and can excite neighboring elements. Thus, waves of excitation can appear. The waves can be plane and spiral. Spiral waves in myocardium emerge only in the case of dangerous arrhythmias and must be treated if they persist. One treatment called low-voltage cardioversion-defibrillation (LVCD) involves periodical applying a small electrical current to an area in the myocardium so that the stimulated zone produces plane waves with greater frequency than the spiral waves. The plane waves begin to occupy broader and broader zones and finally supersede the spiral waves. The aim of the study is to find out the optimal stimulation parameters for the LVCD. Five models with different numbers of parameters are used for the simulation.

The induced drift of spiral waves was simulated with a range of cardiac models, model parameters, stimulation periods, and areas [2]. There were $6{\cdot}8{+}5{\cdot}9$ sets of parameters in total (parameters varied for different models).

As myocardium has fibers along which the excitation spreads faster than across, it considered anisotropic. The wave drift was studied in an anisotropic tissue [3], so fiber direction was a new parameter that also varied. In total, $7 \cdot 5 \cdot 2$ parameter sets were examined.

Another work was devoted to studying LVCD in anisotropic myocardium models with curved fibers using biophysical ionic Luo—Rudy cell model [4]. After measuring the time of the spiral wave drift and determining the type of interaction between waves, the findings were compared with the results of the isotropic and parallel fiber anisotropic cases. In total, seventeen parameter sets were investigated.

The computational program was written in C using a third-party software [5] for parsing INI files.

Although the service was designed for heart modeling problems, it can also be used with another computational clusters and other software.

5 Related Work

To assist domain-specific scientists in conducting numerical experiments on parallel computing systems, various platforms have been created [6]. A number of projects have provided the integration of application software packages with supercomputers. For example, the DiVTB (Distributed Virtual Test Bed) platform provides a task-oriented approach for solving specific classes of problems in computer-aided engineering through resources supplied by grid computing environments. It has a user-friendly graphical interface where parameters of a computational experiment can be specified, and the experiment can then be executed on a supercomputer.

A "Specialized web portal for solving problems on multiprocessor computing systems" [7] is a similar to DiVTB project for remote calculations. The system incorporates several parallel algorithms to solve the inverse gravity of lateral density reconstruction, the structural inverse gravity, and the magnetic problem of the contact surface reconstruction.

A platform called Education-research Integration through Simulation On the Net" (EDISON) [8] has been designed and implemented to access and run various technological computer-aided design software tools. The platform provides an easy-to-use GUI that helps geographically distributed researchers run and share their tools in five areas: computational fluid dynamics, computational chemistry, nanophysics, computer-aided optimal design, and computational structural dynamics.

The Orion system [9] provides a practical and economical interface on the Tianhe-2 supercomputer to enable big data applications to run on Tianhe-2 via a single command or a shell script.

While many systems are limited by an integrated set of algorithms or applications, the distinctive feature of the proposed system is the ability to use various software to run experiments, providing users flexibility and convenience. Also, previous projects did not provide automation, such as launching of a series of computational experiments with varying parameter values. However, the positive experience of management of such systems via a web interface is worth noting and inspired the creation of the proposed system.

6 Conclusion

The paper describes the web-based multiplatform system for launching a series of experiments on a supercomputer. The service interface allows users to set the parameters and run jobs on a supercomputer without working in a command line. The system focuses on minimizing the number of user actions required for launching a large series of experiments. Using the built-in database, a previously conducted experiment can be quickly restarted with new settings. The system

supports INI sections and comments, and allows users to specify a range of values incrementally.

The architecture of interaction among a computing cluster, a database, and an external web application were presented with the algorithm for generating tuples from sets of values with individual parameters, and features of the graphical user interface were also presented. The system is used for carrying out computational experiments to study the drift of spiral waves in myocardium on the *Uran* supercomputer.

The nearest extension of the project is the option of automatically writing to the archive with indexing in the database and the ability to search through previously launched experiments using keywords. Adapting the system for more interactive job control (for example, canceling all the jobs launched during an experiment) will greatly enhance usability. The researchers also plan to add build-in data post-processing methods for convenience.

Acknowledgments

Our study was performed using the *Uran* supercomputer of the Krasovskii Institute of Mathematics and Mechanics.

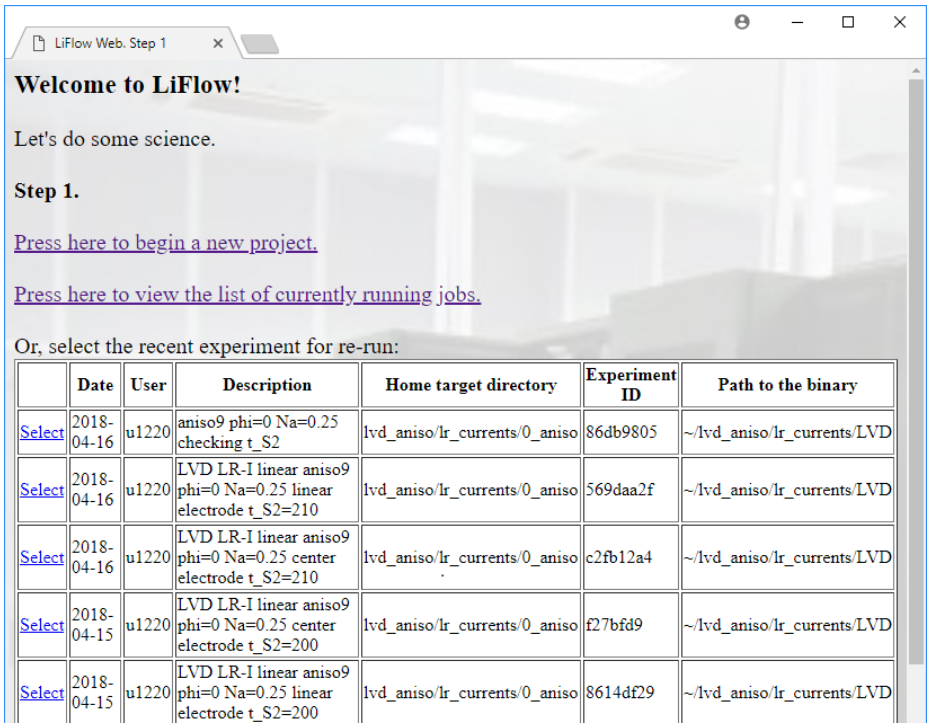
References

1. PostgreSQL: The world's most advanced open source database. https://www.postgresql.org/. Accessed: 2018-06-29.
2. Sergei Pravdin, Timur Nezlobinsky, and Alexander Panfilov. Inducing drift of spiral waves in 2D isotropic model of myocardium by means of an external stimulation. *CEUR Workshop Proceedings*, 1894:268 – 284, 2017. URL `http://ceur-ws.org/Vol-1894/bio6.pdf`. Proceedings of the 48th International Youth School-Conference 'Modern Problems in Mathematics and its Applications', Yekaterinburg, Russia, February 5-11, 2017.
3. Timofei Epanchintsev, Sergei Pravdin, Andrey Sozykin, and Alexander Panfilov. Simulation of overdrive pacing in 2D phenomenological models of anisotropic myocardium. *Procedia Computer Science*, 119:245 – 254, 2017. ISSN 1877-0509. URL `http://www.sciencedirect.com/science/article/pii/S187705091732392X`. 6th International Young Scientist Conference on Computational Science, YSC 2017, 01-03 November 2017, Kotka, Finland.
4. Timofei Epanchintsev, Sergei Pravdin, and Alexander Panfilov. Spiral wave drift induced by high-frequency forcing. Parallel simulation in the Luo–Rudy anisotropic model of cardiac tissue. In *Computational Science – ICCS 2018*, pages 378–391. Springer International Publishing, 2018.
5. N. Devillard. IniParser: stand-alone ini parser library in ANSI C. https://github.com/ndevilla/iniparser, 2016.
6. Stanislav P.Polyakov, Andrey P.Demichev, and Alexander P.Kryukov. Web toolkit for scientific research: State of the art and the prospect for development. *Procedia Computer Science*, 66:429–438, 2015.

7. Elena Akimova, Vladimir Misilov, Aliya Skurydina, and Maxim Martyshko. Specialized web portal for solving problems on multiprocessor computing systems. *CEUR Workshop Proceedings*, 1513:123 – 129, 2015. URL `http://ceur-ws.org/Vol-1513/paper-12.pdf`. Proceedings of the 1st Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists, Yekaterinburg, Russia, Nov. 17th, 2015.
8. Suh Young-Kyoon, Hoon Ryu, Hanki Kim, and Kum Won Cho. EDISON: A web-based hpc simulation execution framework for large-scale scientific computing software. pages 608 – 612, 2016. Proceeding of IEEE/ACM 16th International Symposium on Cluster Cloud and Grid Computing (CCGrid 2016), May 2016.
9. Xi Yang, Chengkun Wu, Kai Lu, Lin Fang, Yong Zhang, Shengkang Li, Guixin Guo, and YunFei Du. An interface for biomedical big data processing on the Tianhe-2 supercomputer. *Molecules*, 22(12), 2017. URL `http://www.mdpi.com/1420-3049/22/12/2116/html`.

Appendix

Typical screens from the web interface are listed below.



	Date	User	Description	Home target directory	Experiment ID	Path to the binary
Select	2018-04-16	u1220	aniso9 phi=0 Na=0.25 checking t_S2	lvd_aniso/lr_currents/0_aniso	86db9805	~/lvd_aniso/lr_currents/LVD
Select	2018-04-16	u1220	LVD LR-I linear aniso9 phi=0 Na=0.25 linear electrode t_S2=210	lvd_aniso/lr_currents/0_aniso	569daa2f	~/lvd_aniso/lr_currents/LVD
Select	2018-04-16	u1220	LVD LR-I linear aniso9 phi=0 Na=0.25 center electrode t_S2=210	lvd_aniso/lr_currents/0_aniso	c2fb12a4	~/lvd_aniso/lr_currents/LVD
Select	2018-04-15	u1220	LVD LR-I linear aniso9 phi=0 Na=0.25 center electrode t_S2=200	lvd_aniso/lr_currents/0_aniso	f27bfd9	~/lvd_aniso/lr_currents/LVD
Select	2018-04-15	u1220	LVD LR-I linear aniso9 phi=0 Na=0.25 linear electrode t_S2=200	lvd_aniso/lr_currents/0_aniso	8614df29	~/lvd_aniso/lr_currents/LVD

Fig. 2. GUI. Step 1

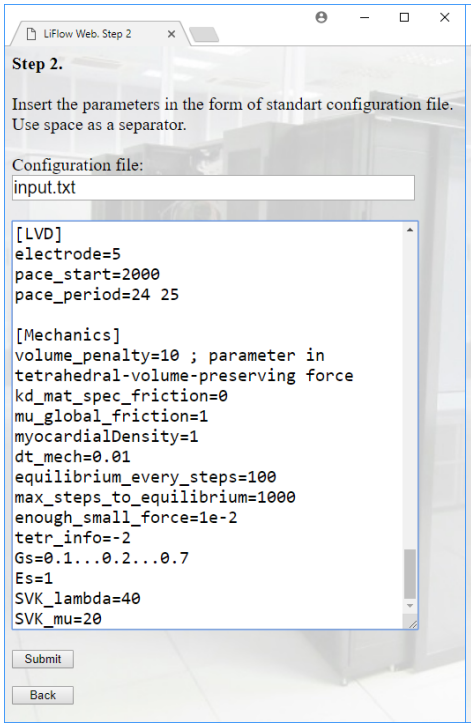


Fig. 3. GUI. Step 2

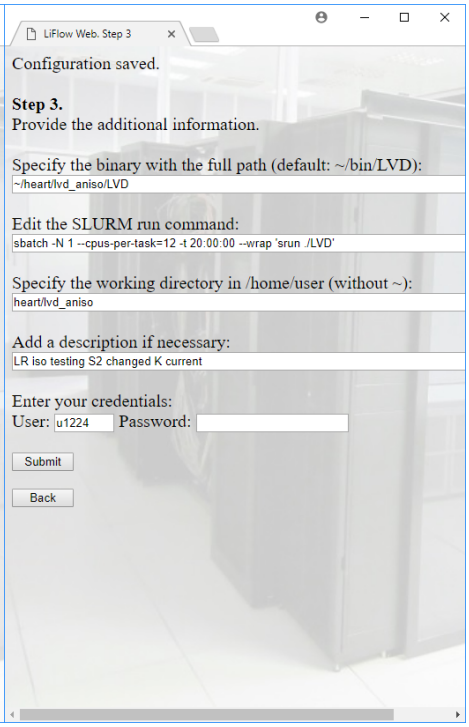
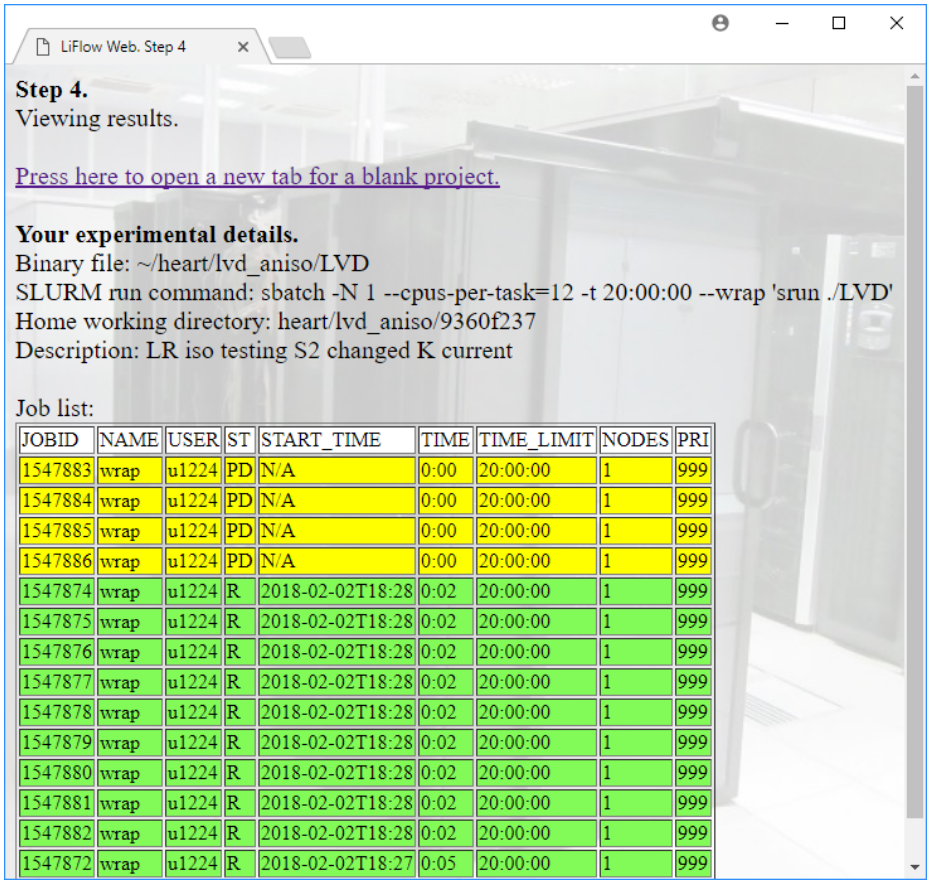


Fig. 4. GUI. Step 3



JOBID	NAME	USER	ST	START_TIME	TIME	TIME_LIMIT	NODES	PRI
1547883	wrap	u1224	PD	N/A	0:00	20:00:00	1	999
1547884	wrap	u1224	PD	N/A	0:00	20:00:00	1	999
1547885	wrap	u1224	PD	N/A	0:00	20:00:00	1	999
1547886	wrap	u1224	PD	N/A	0:00	20:00:00	1	999
1547874	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547875	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547876	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547877	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547878	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547879	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547880	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547881	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547882	wrap	u1224	R	2018-02-02T18:28	0:02	20:00:00	1	999
1547872	wrap	u1224	R	2018-02-02T18:27	0:05	20:00:00	1	999

Fig. 5. GUI. Step 4