



Eugeniy V. Biryaltsev, Marat R. Galimov, Denis E. Demidov,
Aleksandr M. Elizarov

The platform approach to research and development using high-performance computing

ABSTRACT. In this paper, we analyze the prerequisites and substantiate the relevance for creating an open Internet platform that employs big data technologies, highperformance computing, and multilateral markets in a unified way. Conceived as an ecosystem for the development and use of applied software (including in the field of design and scientific research), the platform should reduce time/costs and improve the quality of software development for solving analytical problems arising in industrial enterprises, scientific research organizations, state bodies and private individuals. The article presents a working prototype of the platform using supercomputer technologies and desktop virtualization systems.

Key words and phrases: Internet platform, digital platform, software platform, cloud platform, cloud services, application software, software development, workflow automation software, business automation, software as a service.

2010 *Mathematics Subject Classification:* 65Y05; 68U99,

Introduction

Generally, industry-specific software is developed as an automated workstation (AWS) which comprises similar functionality. Each AWS has its own data storage and GUI. Legal access to the workstation was carried out by acquiring licenses, as a rule, based on the number of users using the workstation (named or simultaneously working)

This study was supported by the Russian Foundation for Basic Research as part of scientific projects Nos. 18-07-00964 and 18-47-160010 and the Tatarstan Academy of Sciences.

© E. V. BIRYALTSEV⁽¹⁾ M. R. GALIMOV⁽²⁾ D. E. DEMIDOV⁽³⁾ A. M. ELIZAROV⁽⁴⁾ 2019

© INSTITUTE OF APPLIED RESEARCH⁽¹⁾ 2019

© LLC “GRADIENT TECHNOLOGY”⁽²⁾ 2019

© JOINT SUPERCOMPUTER CENTER OF RAS^(3,4) 2019

© KAZAN FEDERAL UNIVERSITY^(3,4) 2019

© PROGRAM SYSTEMS: THEORY AND APPLICATIONS (DESIGN), 2019

 10.25209/2079-3316-2019-10-2-93-119



Several AWS of the same company can form a system for processing a single data field. For the AWS of different manufacturers, the interaction is based on integration procedures such as uploading and downloading the data in specialized formats. The software is usually upgraded or modified according to the scheduled one-year releases. Some manufacturers (for example, Ocean Partner Program of Schlumberger [?1]) allow the third-party developers to expand the software with custom modules under their supervision. Generally, an AWS contains many specialized tools, and a user can master them only by taking courses arranged by the manufacturer. One AWS license costs from several dozens to several hundreds of thousands of US dollars.

The current architecture and business models are focused on large corporations which can invest significant funds in the software and personnel, as well as on the relative immutability of the subject field methods. It is also implied that the enterprise business processes are relatively uniform: various AWS of several large software manufacturers covers the marginal differences in these processes. In case of the oil and gas industry, multinational oil and gas companies are the primary consumers of software, and several major manufacturers such as Schlumberger, CGG, ROXAR, and Paradigm [?2, ?3] share the software market.

Expectedly, such a software-developing practice is typical for the industrial society. Transition to a post-industrial economy implies a series of new issues which correlate poorly with the architecture and business model stated above. In particular,

- Hypersegmentation of markets up to customization and personalization for each consumer: saturated market forces manufacturers to focus on narrow fields. Therefore, industry-wide software becomes both redundant and insufficient at the same time. Customization and maintenance of large software systems is too expensive;
- The growing role of cross-industry and cross-cutting projects, involving the use of functionality from different subject areas within the framework of a single project; the construction of information transfer systems between large systems with complex databases, the integration of which becomes a complex and error-prone process.
- Integration of the development of innovative companies that focus, as a rule, on solving a narrow range of tasks. The developed software product requires integration with existing software systems, which is currently achieved by industry sales of a large company's startup. The inertia of this process, as well as the fact that large companies

often buy startups to avoid competition with their own, less perfect development, impede technological advancement;

- Intensive use of computational models of objects and processes in the industry, including multiphysics models which require modification of algorithms according to the subject-specific issues, a dimension of the problem, accuracy and adequacy of the solution;
- The emergence of the data-driven approach which implies the study of accumulated data directly in the production cycle, the construction of numerical models based on these studies, and their immediate application to obtain competitive advantages.

Therefore, a modern industry project implies individual researches, construction of specialized models based on this data, as well as the development of rational technologies based on these models. The software which implements this process shall be created dynamically in the course of project development. On each stage, it shall be possible to apply different methods of analysis, modeling, and synthesis from various subject fields. Expectedly it is tough to solve this task basing on industry-specific automated workstations because of technological, evaluating and financial reasons even for large companies. It comes practically impossible for small and medium-sized business. The expenses of searching for the suitable software, hiring or training personnel, coordinating integration and modifications of heterogeneous software are very high, even on software leasing basis (SaaS model) or through the use of free demo versions. Separate issues are accounting and registration of intellectual property, such as algorithms and data sets.

We should note that most of the expenses mentioned above are the so-called transaction costs [74]. Transaction costs played a small part compared to the transformation costs of production in the pre-industrial society, therefore industrial technologies were aimed at transformation costs reduction. Transformation costs reduced tenfold, and transaction costs have become a significant part of the product value through the development of industrial technologies. This is especially noticeable in complex multilateral markets, as well as in markets with information asymmetries, when market participants are completely differently informed about the properties of the product they are interested in. New industry projects, including research, design, and development are often cross-cutting and interdisciplinary, are a very complex object for a transaction, the transaction costs of which account for the overwhelming share.

Don Tapscott's [?5] digital economy concept responds to an increasing share of transaction costs in the final product value. Within the framework of this concept have been developed and partially implemented both general methods, such as open innovations [?6], multilateral Internet platforms [?7], blockchain technologies, flexible project management technologies, and specific techniques for software development: cloud technologies, microservice architecture, continuous delivery, graph models, etc.

This paper describes an approach to software architecture development based on digital economy methods, which minimizes the transaction costs, as well as business models for the implementation of complex industrial projects. A software prototype based on the following approaches and principles is also presented herein.

1. Approaches

Software enhancement acceleration. Solving computationally intensive applied problems for which no universal solution algorithm is known in advance requires fast software modification. The well-known problem of software upgrade cost reduction is caused by accelerating changes in the business requirements, business processes improvement, and information systems integration. The adopted solution is to use microservice architecture, dataflow computation models, and scripting languages (similar software architecture specified in [?8]), which allow changing the program code in the process. The methods mentioned are widely applied in such fields as data mining and information systems integration. We may consider any project as a set of operations on data extraction, conversion, and data analysis. Therefore software implementation models for data mining and ETL procedures may be applied to a whole project.

Reducing communication costs. Communication among different parties is the sphere of great interest for Internet platforms [?7]. Internet platforms for optimization of communication on a specific topic of interest (social networks), shopping (online stores), transport booking (Uber, Yandex-taxi), accessing video content (Netflix, YouTube) and some other activities are developing rapidly. Also, there are social networks aimed at business problems solution, such as business contacts establishment (LinkedIn) and collaborative software development (GitHub); however, they reduce the communication costs for one of the parties only. It would be useful to integrate the search for partners and resources with the coordination of

activities on a single platform and combine them with the possibility to execute software components from the communication environment directly. This approach avoids uploading and downloading of the software and data used in the project, transferring them among the users using external software and allows to control project implementation according to the actual stages of data processing inside the platform.

Minimization of intellectual property use cost. Protection and accounting of the intellectual property use are serious constraints for the implementation of research methods in the applied industries as well as the integration of the already known methods. Protection of the intellectual property by signing individual agreements with each user makes the time required for accessing the information resources (IR) unacceptable. Whereas, the integration of specific methods into sophisticated automated workstations and systems makes it unaffordable to use these methods in the research process. The solution to this problem today might be the extensive use of open licenses [76], which allow using intellectual property based on general rules, avoiding signing individual agreements, and information resources leasing, known as SaaS.

Another problem of IR management - accounting for the use of individual modules - is removed by the methods of distributed registries [9] (blockchains), which allow recording any transactions without trusted registration centers. It is vital that reducing the cost of verifying the use of intellectual property allows you to take into account the micro-use of individual modules and makes it inappropriate to bundle application software solutions into large application software packages. Accounting in distributed registries for the use by all participants of the intellectual property platform makes it possible to organize billing and mutual settlements trusted by all participants.

2. Prototypes

The platform is set to provide users with multiple multidirectional features such as:

- direct acyclic graph representation of an algorithm;
- dynamic control of algorithms and processing flows;
- data visualization;
- an opportunity to design specialized graphical user interfaces for algorithm management;

- team collaboration support;
- support for distributed and high-performance computing;
- support for data version control;
- an opportunity to process and store vast amounts of data;
- support for computing infrastructure management;
- keeping track of different algorithms and computing resources usage.

At the moment, a full-fledged software platform, which would already have all the necessary properties, is not available on the software market. At the same time, several companies offer software products that implement some of the platform features mentioned above. Therefore, such products can be considered possible prototypes. Since there are many functions required to implement, then there are quite a lot of potential prototypes, respectively. We can conditionally divide such prototypes into the following groups:

- software tools for big data analysis and scientific research support;
- process automation software frameworks;
- business process modeling tools;
- integration platforms;
- open Internet platforms for software developers.

Software tools for big data analysis and scientific research support.

The software products which implement the environments for analytics, big data processing, and building machine learning models might be the closest prototypes to the platform we offer. Such software is required to possess the maximum flexibility and variability of the implemented algorithms and to allow the end-users to modify the data processing pipeline. Following these requirements, the software comes as a set of data processing algorithms, means of graphical visualization of both the input data and the processed results, as well as a visual tool for building a dataflow diagram. Presently, a range of such software products is known: Orange, RapidMiner Studio [?10, ?11], Knime Analytics Platform [?12, ?13], EasyMorph, and AdvancedMiner. These software products solve, among others, such tasks as data preprocessing, feature selection, clustering, classification, regression analysis. In other words, they mainly focus on mathematical and statistical analysis. Such software provides an integrated environment for scientific research; allows implementing a full cycle of data analysis which comprises reading data from different sources, data transformation, cleaning, the analysis itself, visualization and export

of the data. Unfortunately, the mentioned software products, although promising, have the following conceptual disadvantages in the context of the considered platform:

- the lack of capabilities to control versioning of data and processing graphs;
- user interaction with the system obeys direct work with the processing graph; there is no possibility of designing a specialized user interface;
- support for distributed computing or hosting in the cloud is mostly limited by commercial versions of software;
- data storage organization is considered beyond the scope of system functionality;
- the lack of opportunities for collaborative work on a processing graph.

Process automation software frameworks. The use of flowcharts or workflows is one of the most important specific features of the platform. This feature should not only present in graphic diagrams but should be inherent in basic software modules if the platform, its architecture and development principles.

At present, software development tools (libraries) for low-level data workflow implementation are being intensively developed. These tools allow performing development, planning, and monitoring of data workflows in time in the form of flowcharts. Apache Airflow, Luigi, Nextflow, and many others are notable examples of such software.

Such software tools oriented on use by professional programmers are not fully-featured software products. Descriptions of processing algorithms are directly coded either in a corresponding programming language or in the specialized files. Non-professional end-users are mostly unable to use and customize the descriptions. Additionally, there are several shortcomings:

- visual workflow designer is not implemented, a flowchart is generated based on a specialized script;
- stages of flowchart development and execution are isolated.

Business process modeling tools. The principles of breaking down a long and complex process into multiple steps are actively used in business process modeling (BPM). Currently, there is well-developed methodological support, as well as appropriate modeling languages and software development tools.

Well-known software products are Activiti BPM Platform and Camunda. Similar software tools are also provided by integration platforms

manufacturers in such products as JBoss, Websphere. These software products are offered both as a process flowchart development tools and as an environment for their execution and monitoring; it is also possible to build simple user interfaces to interact with the system.

The described group of software products also looked very promising as the main prototype for the developed platform, however, after some experiments, the following shortcomings were revealed:

- no support for high-performance computing;
- restricted changes in the flowchart in runtime;
- no possibility to process single data set in multiple ways in separate process nodes;
- no possibility of scientific data complex visualization.

Integration platforms. Integration software (EAI, ESB) is another well-known software class applying the principles of decomposition of complex processes into stages and their representation as flowcharts when developing and monitoring. This software allows users to organize and control the data workflow within the information systems of the company; to achieve that, it widely uses flowchart representation.

The widely known examples of such software are Apache ServiceMix, Mule, JBossESB, IBM WebSphere, WSO2, and Apache NIFI.

In addition to the generic integration platforms, we should mention specialized software for the oil and gas sector, for example, the Whereoil platform [?14] by Kadmi, which claims to provide information support for business processes related to big data management. This product also allows creating visual representations of workflows, linking them with information resources (modeling systems) directly, and organizing end-to-end project management.

Such software products are primarily designed to perform integration procedures and data flow management. In fact, the urgent tasks of complex mathematical data processing, research, and visualization of results are not supported. These tasks are partially supported in the mentioned Kadmi product, but the lack of sufficient information on this proprietary product in the public domain does not allow its evaluation.

Open Internet platforms for software developers. Tools for creating a software development infrastructure online we also consider as prototypes of the developed platform. Historically, the first platforms of the kind provided versioned storage of source codes. Currently, the Github, Gitlab, Bitbucket, and other integrated services for the IT project development

and management provide team management, project documentation, continuous integration tools in addition to code storage. There are Internet platforms that allowing to create an entire development and testing environment, i.e., a set of specialized servers hosting software developed by the users (Platform as a Service). For example, Salesforce offers a cloud-based PaaS platform Heroku [?15, ?16], which makes it possible to both develop and execute programs in the cloud. Amazon allows creating the necessary virtual hardware and software infrastructure quickly, including a high-performance one. The company provides tools for cloud software development, including AWS Step Functions [?17] for the orchestration of the components and microservices of the distributed applications as workflows. SAP offers the HANA platform [?18], which supports distributed data storage, development and data processing tools as well as a runtime environment for providing the user with data. The SAP Data Hub [?19] allows connecting to different data sources and generating data sampling and processing flows.

The described class of software products we consider as a prototype of our platform with regard to providing multiple tools for computing infrastructure and software development management. However, there are significant disadvantages:

- these products are focused on the community of professional programmers;
- the software is primarily intended for developing applications with a web interface;
- they are focused on software development and operation rather than data analysis and processing;
- the full-featured platforms (products from SAP, Amazon and Salesforce) do not provide permanent free access or are very expensive.

Two new projects, which aim to create a commercial distributed network to solve tasks related to machine learning, Pallium.network [?20] and SingularityNET [?21], look very promising. They assume that third-party users will be able to connect their software as separate network nodes and then offer them for commercial use to other members. They propose to use blockchain technology to protect intellectual property and pay for resources used. However, such software products are under development and can be used mostly for methodological reference.

Despite the great diversity of available software products, each of the mentioned software groups has many significant drawbacks. They

prevent direct use of these products in the implementation of the platform according to our concept. Thus, although initially, we were willing to modify the existing software according to the platform requirements and thereby reduce the time required for the development of a full-featured platform, we had to start developing a new software product, the prototype of which we present below.

3. Proposed approach

The proposed solution is based on the approaches to the software architecture and business architecture design, as well as on the intellectual property management principles.

Software. We propose an Internet platform comprising a central application server (repository) that provides storage of algorithms and their components, user registration and management, communication services, as well as billing for the use of algorithms and their components.

The algorithms and their components are executed in a runtime environment (engine) which is installed either on the user's host or on a virtual machine in a cloud cluster when a user joins the platform.

The engine allows creating, customizing, and using the algorithms to solve specific tasks. User interaction is carried out through either web interface or desktop client installed on local or virtual machines.

User roles. There are three main user roles on the platform:

- experts who develop new algorithms within the scope of grants or contract works, or on the initiative basis; the developed algorithms are supplied with identifying information and then deposited in the platform repository;
- advanced users who select and adapt the existing algorithms to the new or modified tasks;
- users who solve specific tasks using the algorithms developed by experts and specialists; they interact with the system as if it was an ordinary software.

License policy and user interaction. The license policy in the field of the intellectual property stands based on accession to the license policy of the platform (accession agreement). The agreement should contain the obligations of the user to comply with the copyright of the third parties—participants of the platform (including property rights when using the external intellectual property), as well as the terms of service.

The prior payment scheme for algorithm use is royalties; however, for exclusive algorithms, there is an option to use custom work contract with a lump-sum payment. The mechanisms of distribution of the remuneration for derivative works applies to the modified algorithms. We suppose the official remuneration for contractors.

4. Advantages of the proposed approach

We expect a significant reduction of time and cost required for the software modification. The identification of software-technical and algorithmic problems in the early stages improves the quality of solutions. When professional users get the opportunity to independently change data processing algorithms and the graphical interface that controls the algorithms, the need for professional programmers disappears. System design and experimental operation merge into a single iterative process.

To illustrate, we introduce our firsthand experience in such software development process. One of our latest projects was a software system for data processing and analysis using a specialized proprietary geophysical method. The peculiarity was that the method needs a careful adjustment to specific geological conditions at hand. We had been developing the software for a year and a half; however, the result could not be considered satisfactory.

Even though the product was created and had extensive functionality, its deployment was postponed many times, because each time the adjustment of the algorithms was required. Besides, it took quite a lot of time for programmers to implement applied algorithms: they had to understand the task, analyze the algorithm, and use the programming languages that are often unfit for scientific algorithms. Finally, we decided to develop a new product from scratch based on the principles described above. As a result, the new program (alpha-version of the prototype described below) became released only six months later. Thus, we witnessed the product implementation speed at least tripled, and the cost of the professional programmers' service diminished correspondingly.

The proposed solution has additional advantages listed below.

Task performance management. A DAG (directed acyclic graph) forms the object that implements a business process (analytical research, project development, reporting) is also a network schedule. As soon as the intermediate data structures are filled, it is possible to track the work progress, estimate the timing and performers workload. Unlike standard

software architecture, DAG-architecture does not require additional integration with project management systems. Project and resource management systems can appear into the platform as additional services.

Verification of complex results. According to the proposed approach, the performed analytical research, report, and project contain the initial, final, and intermediate data as well as algorithms for obtaining them. Blockchain markup for data and tracking algorithms ensures consistent results. It is possible to rerun the DAG object from the initial data to confirm the final result.

Self-descriptiveness. Complex information systems with a large number of functions distributed over automated workstations are extremely problematic to master. A DAG is self-evident to a specialist as a sequence of business functions performed for a specific project.

Integration of applied research. The development of a new mathematical model or business function with the help of third-party experts is currently a process with enormous transaction costs, its practical implementation takes a long time. Using of Internet platform allows significantly accelerate this process and bring the applied research closer to solving the industrial tasks. The applied research may be initiated by formulating the requirements to the final result and providing the contractor with access to the test data. The result of the applied research is a DAG object that demonstrates the achieved results on the test data and is ready to be used on the real data.

Applicability to cloud technology. The proposed architecture is based, in particular, on the approaches developed for distributed computing. Therefore it is completely ready for operation on public or private clusters.

5. Prototype

Consider the overall architecture of our platform prototype, implementing the described concept (??):

- central registration node (CENTRAL REG NODE);
- controlling cluster (CONTROL NODE);
- computing cluster (COMPUTE CLUSTER);
- client-side applications (Cloudviewer).

Central registration node is a single aggregation point for controlling and computing clusters designed to provide integration processes. The registration node also provides user management for all clusters connected to the platform. Also, it stores all shared objects.

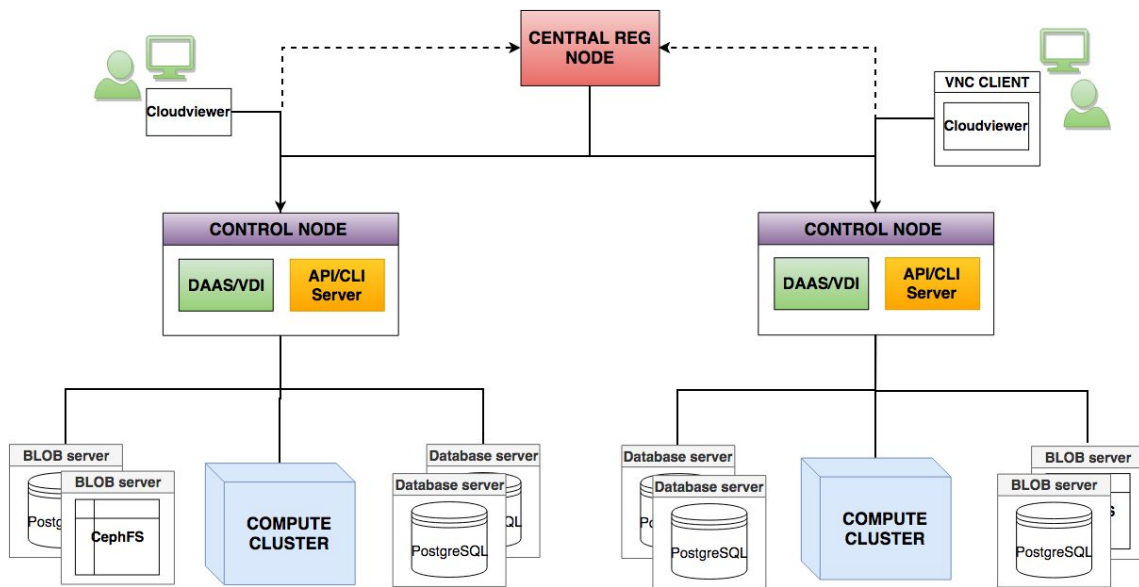


FIGURE 1. Platform architecture

Controlling node is a hardware and software infrastructure that implements basic platform services and provides remote access to the platform on the Internet. Resource-consuming programs are executed on the connected computing clusters.

Computing cluster is a high-performance hardware and software infrastructure designed to execute resource-consuming user programs (scripts). Compute clusters can be both integrated into the controlling nodes and the external clusters within the public network.

Client-side applications are a set of software that allows the end-user to interact with the services of the platform's controlling node. The applications can connect to any controlling nodes they have access to. A central registration node can be used to find available controlling nodes.

The platform architecture provides means for flexible customization and makes it possible to deploy the platform both in private groups network of a single company and on the Internet for public use. There exists a technical possibility to integrate several groups into a bigger one, which is often required when several contributors work on the same project.

The controlling node is the main element of the platform and consists of:

- the central application server;
- the relational database;
- the object database;
- services.

The central application server is a single access point to the services provided by the controlling node. It also provides:

- API & CLI;
- platform services hosting/registration;
- user interaction with the platform.

The relational database is intended to store platform service data as well as the structure and metadata of flowcharts, data and metadata of platform services, and lightweight datasets.

The object database is used to store large datasets. The data is stored directly in a specialized key-value archive which is a PostgreSQL 10 database in our case. In the table records, the datasets are represented by a byte array (bytea). To reduce the table load, the data is distributed across a group of tables according to the first characters of the storage hash (key). It is also possible to split the data into blocks and store those blocks as a chain.

Services are software modules deployed on a central application server which implement the functionality of the platform controlling node. The services interact with the main modules of the platform controlling and computing clusters and allow the user to access the platform functionality.

Thus, the controlling node allows storing and processing datasets of different types and sizes as well as team collaboration.

A vital platform feature is the ability to carry out resource-intensive data processing and simulation tasks. The integration of the prototype controlling cluster with a high-performance GPU and a VDI system implements this principle. We described their hardware and software architecture in [?22]. We use SLURM resource and task management system to conduct distributed computing, and NQ management system to perform local tasks.

As part of the prototype, we implemented a number of client-side applications which provide means for:

- administration of the platform;
- user personal account;
- project management (project editor and project manager);
- team collaboration based on the forums and galleries where users can communicate and exchange flowcharts and datasets.

All the client-side applications are desktop applications designed using Python3 programming language and PyQt5 technology. In this regard, all the user activities connected with the processing of large datasets are executed on the VDI cluster. We have started to develop a web client for using platform services.

There is a possibility to expand the client-side application functionality by third-party users provided by the plugin system. The plugin allows both changing the graphical interface of the application and expanding its functionality.

One of the Russian geophysical companies is already widely using the prototype to implement current projects in the oil and gas industry.

6. Advantages of the prototype

The prototype can be considered only as an intermediate version of the platform. At the same time, it already has several significant advantages over existing software products described in Section ??.

The prototype is *the first* iteration of deep integration (synthesis) in a single system comprising various tools for team collaboration and data

analysis and processing which were previously implemented as separate software products. Unlike the existing analytics platforms, the prototype does not constrain the users by a specific programming language when developing scripts. If a language does not have native support in the prototype, then the user can independently implement the process of data upload/download into the script from the file system.

Another feature is the possibility to develop a specialized graphical interface (automated workstation) over the flowchart nodes to hide the technical details of flowcharts from the end-users and to simplify their interaction with the program. The GUI (automated workstation) for data input/output and visualization of parameters and data related to specific flowchart nodes distinguishes the prototype from existing data analysis systems such as Knime or Rapidminer.

The current state of the flowchart determines the graphical interface displayed to the user. To create an automated workstation, a special visual designer should be used; it allows composing new GUIs of various input panels and forms for visualization of data from different nodes. In the prototype, a user can create such an automated workstation for a separate node (??), a group of nodes and the whole flowchart.

Therefore, a user can work with GUI rather than the flowchart. It allows creating templates of data processing algorithms as specialized automated workstations for less qualified users. At the same time, an advanced user can easily switch to flowchart mode and edit the necessary scripts and parameters.

The prototype advantages also include flexible control of the algorithm and its parameters versions, as a user can run a flowchart node for multiple times, combining them in different ways. It means a user can analyze every episode of the node execution regardless of changes in the algorithm or dataset made after that. A single sequence of all the flowchart nodes executions allows the user to get a complete history of the flowchart execution at any time (??) and rerun any node from any point if necessary.

All the flowchart nodes executions are connected into a single sequence, so the user can get a complete history of the flowchart execution at any time (??) and rerun any node from any point if necessary. We must note that there are some limitations related to the history of changes in the flowchart itself, but we should overcome them in the future versions. Such a possibility of a retrospective analysis of the research process is extremely useful in both scientific and business tasks and absent in other platforms studied.

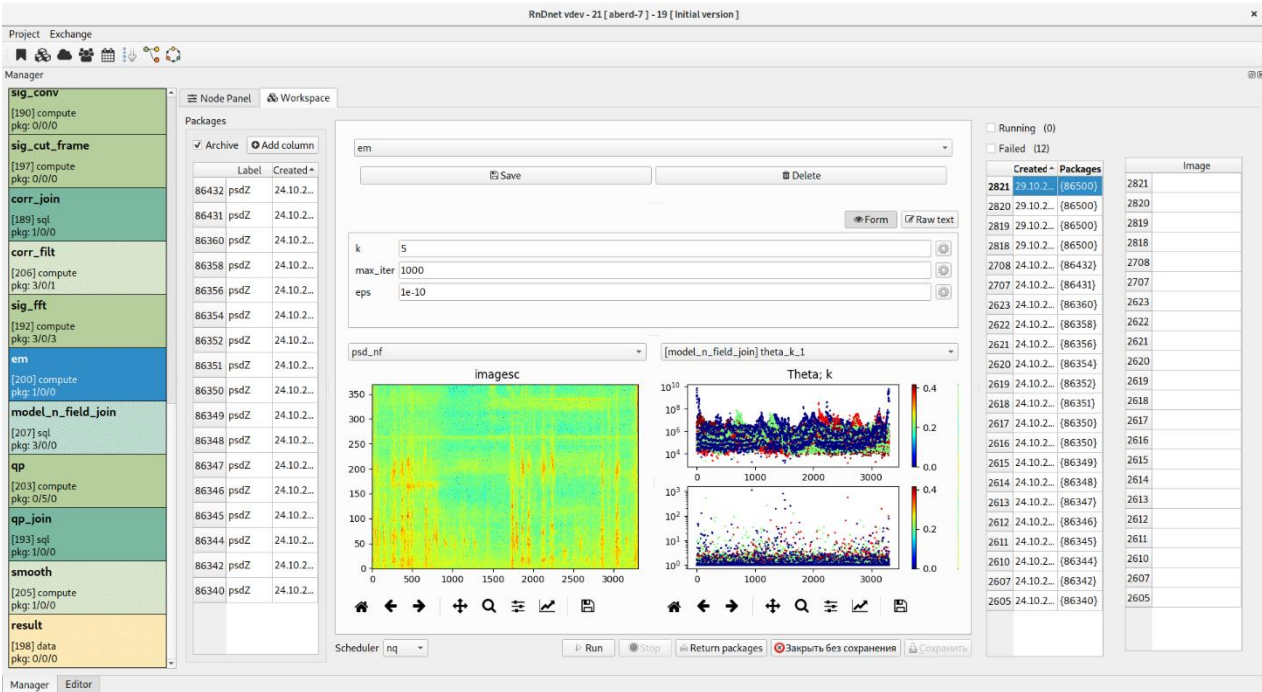


FIGURE 2. Automated workstation sample

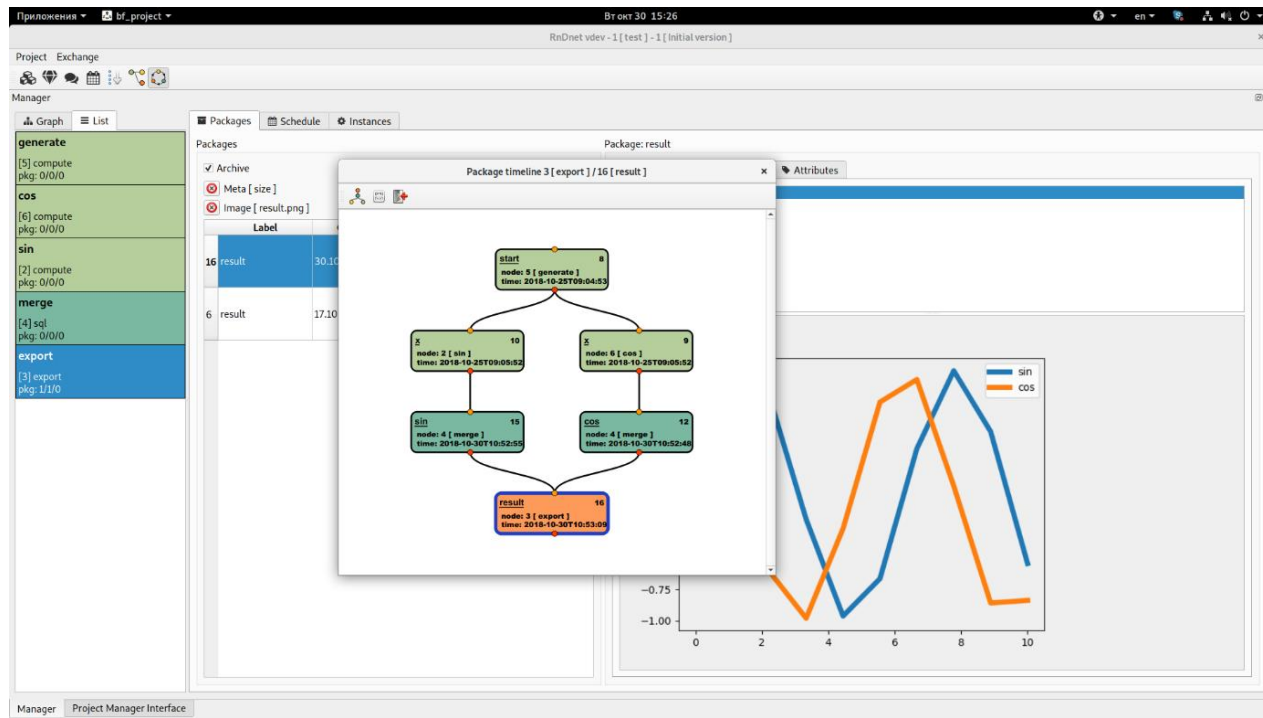


FIGURE 3. Example of the processing timeline

In the early analogs of the created platform analyzed above, team collaboration is available only in commercial versions of the software. It sharply limits the possibility to apply these products to complex scientific or business tasks involving many participants.

Our prototype initially aimed for collaborative work on projects, including problems where the data and computing clusters are geographically dispersed. Technically, each algorithm (flowchart node) can run on different computing clusters or users' machines. We anticipate that instances of the platform prototype installed by different users and organizations will join into a shared network for algorithms exchange and data transfer from one flowchart to another. The prototype has specialized export/import modules for this purpose.

A similar feature is announced by Pallium.network and SingularityNET projects; however, both of them are supposed to build a single computing network. In contrast, our prototype of the software platform allows building several independent computing networks.

Forums built into the prototype provide a promising functionality (previously not met by the authors) of exchanging parts of the processing graph along with the necessary data between different users. User directly can upload the nodes of the flowchart and datasets on the forum (??) from one project and download them into any other project. It allows the users to discuss production or scientific topics and exchange modules directly, which helps to check them by merely uploading these modules into a new project. In many cases, it will solve the problem of error reproduction which is well-known to technical support, since the user will be able to submit not only the description of the problem but also the required datasets and flowchart nodes.


Currently, the capability of project management within the platform is at the initial stage of development. One of the upcoming features is an ability to assign the executor (algorithm developer) for the specific node, with the ability to manage task timeline - a deadline and a time the development has started (??). Thus, it will be possible for a company to control the process of research implemented by its employees. In the future, the project manager shall be informed on project status update (nodes added, nodes successfully tested) to make research more manageable and transparent. Additionally, the platform user can generate a research report, it requires only to select and visualize the necessary datasets and to enter additional information (??).

Forum Browser

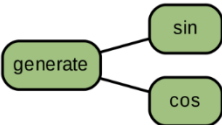
Nothing works here!

Clouds
tp
tesla1
Forums
Generic for...
bir

Message

 **Login:** demidov
Name: Denis Demidov
email: dennis.demidov@gmail.com
cp nov6. 14 09:59:49 2018


How to make this work?



```
graph LR; generate[generate] --> sin[sin]; generate --> cos[cos];
```

Login: gradient
Name:
email:
BT nov6. 27 15:01:53 2018

You can use this nodes.



```
graph LR; generate[generate] --> sin[sin]; generate --> cos[cos]; sin --> merge[merge]; cos --> merge; merge --> export[export];
```

◀ Back

Add comment Get attachment Refresh

FIGURE 4. Forum with attached flowcharts

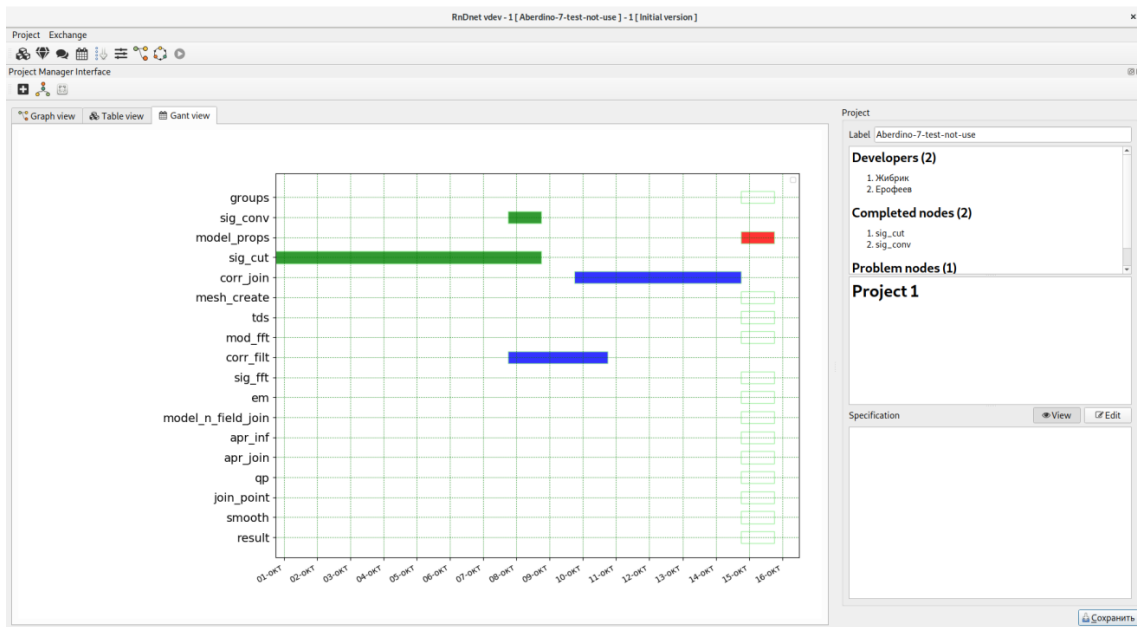


FIGURE 5. Gantt chart

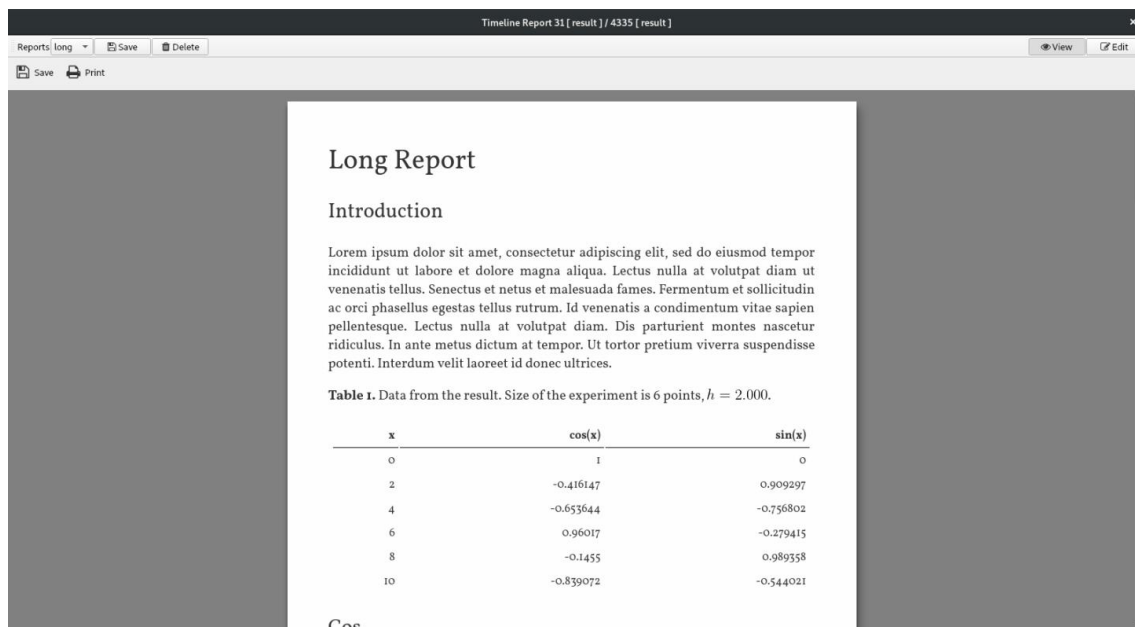


FIGURE 6. Example of a report

We described the most exciting features of the software platform prototype, each of them is not unique; however, in the described combination, they have not been implemented before in the existing software. Ultimately, this synthesis will lead to a breakthrough software emerged in the market of analytical data processing in the production and scientific fields.

7. Plans for further development

The prototype presented in this paper has demonstrated the feasibility of the proposed approach, however, there is a number of technological features not yet implemented such as the interoperability, choosing a strategy for working with versions in a multi-user environment, ensuring data security as well as improving the usability of its user interface.

In our case, we consider interoperability as the need to configure the software and system environment of computing nodes to be able to execute user scripts (programs). This process is not automated yet so the user must either request system administrators in advance for the necessary settings or create the appropriate Python virtual environment or Docker container on his own. Apparently, for a large number of platform users, it is necessary to automate this process. For example, users of the Heroku platform customize the computing infrastructure by choosing the necessary software in a special visual designer; alternatively, in AWS, each executable script should be provided with an additional file containing a description of the necessary infrastructure. Our immediate task is to choose one of these two strategies.

Some technical and conceptual issues emerge when multiple users work on the same project. The system works as a single dataflow, and the flowchart is available for all project participants, which means the changes made by one user immediately affect other users of the project. Such straightforward behavior can cause problems when the user needs to conduct some experiments with a sequence of computing nodes without disrupting the current business process. Currently, to do that it is necessary to clone the project along with the datasets, which is inconvenient in some cases. An alternative solution is to use the versions of the flowchart and dataflow. We have developed a prototype version implementing this approach. However, we did not consider it in this article due to insufficient testing.

Also, there are still issues related to protecting the proprietary data processing algorithms, which are scripts written in interpreted languages from unauthorized copying and use. The methods of software protection

known so far do not guarantee complete safety. A possible solution is to make the algorithm physically unavailable to the user, that is to conduct the computing on the isolated computing nodes as well as to forbid executing the algorithm in interactive mode. A possible data leak makes it very difficult to use cloud computing with sensitive data. This problem is now reflected as the concept of safe computing, fundamentally excluding the possibility of leakage. However, such methods are at the initial stage of development; therefore, they cannot be used for industrial purposes yet.












Another problem is related to building a sophisticated automated workstation for a non-professional user to control the flowchart. Experience shows that standard graphical components are not always suitable for the end-users. They need to develop specific graphical components to make the work more convenient. Currently, the prototype supports developing new components but this requires to involve professional programmers which, in general, reduces the speed of platform development. Thus, our immediate objectives are to develop a broader range of graphical components for control and visualization as well as to create a specialized visual designer that allows the user to create new components on their own. We have not mentioned the organizational and financial arrangements on the proposed platform. The platform approach itself implies a major change in the structure of scientific projects transaction costs which will lead to the changes in both process participants roles and their arrangements, including financial ones. Likely, modern financial technologies (FinTech), including blockchain, smart-contracts, and cryptocurrency can be used to optimize those organizational and financial arrangements. Perhaps the problem of unauthorized access can be solved by transferring it from the technical level to the organizational and legal level by registering the actions of all platform participants and analyzing user activity using data mining methods. We hope to answer these questions not only theoretically, but also while applying the platform to the different industries on an experimental basis.










Conclusion

The problems related to the applied computing problems which are described, assigned and discussed above can be solved by using a sophisticated software platform which includes tools for creating software based on graph-modular architecture as well as communication tools. The platform allows executing flowcharts, generating tools for intellectual property registration, and billing using blockchain methods.

The proposed approach can be used to develop software for various areas of industry and public administration within the digital economy paradigm. In particular, the development of situation centers of the Russian Federation region heads based on the proposed digital platform looks very promising [723]. Such a platform will help not only in the creation and development of these situation centers but also in laying the groundwork for a completely new type of interaction between the state, business, and citizens while solving problems the society is facing. *Acknowledgments.* The authors are grateful to CJSC “Gradient” for the invaluable contribution in the development of the software system.

References

- [1] *Ocean software development framework* (access date 11.02.2019).  ↑
- [2] *Market research of geophysical software for oil and gas industry*, O2Consulting, M., 2014 (access date 11.02.2019) (In Russian).  ↑
- [3] B. G. Levin. “Geoplat Pro” — the software platform for the oil-field exploration and development problems (access date 11.02.2019) (In Russian).  ↑
- [4] R. H. Coase. “The nature of the firm”, *Economica*, **4**:16 (1937), pp. 386–405.  ↑
- [5] D. Tapscott. *The digital economy: promise and peril in the age of networked intelligence*, McGraw-Hill, 1995, 342 pp. ↑
- [6] H. W. Chesbrough. *Open innovation: the new imperative for creating and profiting from technology*, Harvard Business School Publishing, Cambridge, MA, 2003, 227 pp. ↑
- [7] G. G. Parker, M. W. Van Alstyne, S. P. Choudary. *Platform revolution: how networked markets are transforming the economy and how to make them work for you*, W. W. Norton & Company, 2016, 352 pp. ↑
- [8] E. V. Biryal'tsev, M. R. Galimov, A. M. Elizarov. “Workflow-based internet platform for mass supercomputing”, *Lobachevskii Journal of Mathematics*, **39**:5 (2018), pp. 647–654.  ↑
- [9] S. Nakamoto. *Bitcoin: a peer-to-peer electronic cash system*.  ↑
- [10] RapidMiner (access date 11.02.2019).  ↑
- [11] Z. Prekopcsák, G. Makrai, T. Henk, C. Gáspár-Papanek. “Radoop: analyzing Big Data with RapidMiner and Hadoop”, *Proceedings of the 2nd RapidMiner Community Meeting and Conference*, RCOMM 2011, 2011, pp. 1–12.  ↑
- [12] KNIME Analytics Platform (access date 11.02.2019).  ↑
- [13] W. A. Warr. “Scientific workflow systems: Pipeline Pilot and KNIME”, *Journal of Computer-aided Molecular Design*, **26**:7 (2012), pp. 801–804.  ↑
- [14] Whereoil (access date 11.02.2019).  ↑

- [15] Heroku (access date 11.02.2019).  [↑](#)
- [16] N. Middleton, R. Schneeman. *Heroku: up and running: effortless application deployment and scaling*, O'Reilly Media, Inc., 2013, 100 pp. [↑](#)
- [17] AWS Step Functions (access date 11.02.2019).  [↑](#)
- [18] V. Sikka, F. Färber, A. Goel, W. Lehner. “SAP HANA: the evolution from a modern main-memory data platform to an enterprise application platform”, *Proceedings of the VLDB Endowment*, **6**:11 (2013), pp. 1184–1185.  [↑](#)
- [19] SAP Data Hub (access date 11.02.2019).  [↑](#)
- [20] Pallium Computing Network Concept (access date 11.02.2019).  [↑](#)
- [21] B. Goertzel et al. *SingularityNET: a decentralized, open market and inter-network for AIs*, 2017 (access date 11.02.2019).  [↑](#)
- [22] A. Belyaeva, E. Biryaltsev, M. Galimov, D. Demidov, A. Elizarov, O. Zhibrik. “Architecture of HPC clusters for Oil&Gas Industry”, *Program systems: Theory and applications*, **8**:1 (2017), pp. 151–171 (In Russian).   [↑](#)
- [23] E. V. Biryaltsev, R. N. Minnikhanov. “Situation center of the Russian Federation region head in the digital economy paradigm”, *Current health and safety issues: intelligent transport systems and situation centers*, Materials of the V International research-to-practice conference. V. II, Center for innovative technologies, Kazan, 2018, ISBN 978-5-93962-865-5, pp. 3–11 (In Russian).  [↑](#)

Received 18.12.2018

Revised 24.04.2019


Published 27.06.2019


Recommended by

prof. Sergey M. Abramov


Sample citation of this publication:

Eugeni V. Biryaltsev, Marat R. Galimov, Denis E. Demidov, Aleksandr M. Elizarov. “The platform approach to research and development using high-performance computing”. *Program Systems: Theory and Applications*, 2019, **10**:2(41), pp. 93–119.

 10.25209/2079-3316-2019-10-2-93-119

 http://psta.psir.ru/read/psta2019_2_93-119.pdf

The same article in Russian:

 10.25209/2079-3316-2019-10-2-121-153

About the authors:**Eugeni Vasiljevich Biryaltsev**

Expert in the field of specialized information systems, Candidate of Engineering Sciences. Author of more than 50 publications (including 3 software copyright registration certificates and 2 inventions). The general director of LLC “Gradient Technology” (SKOLKOVO resident). Head of the Digital Technologies Center at the Institute of Applied Research of the Tatarstan Academy of Sciences.



0000-0002-5193-8627

e-mail: Igenbir@yandex.ru**Marat Razifovich Galimov**

Software development expert for the oil and gas industry, Candidate of Engineering Sciences. Assistant director of LLC “Gradient Technology” (SKOLKOVO resident).



0000-0002-4997-6878

e-mail: glmvmrt@gmail.com**Denis Evgenievich Demidov**

Expert in high-performance computing using GPGPU technologies, Candidate of Physical and Mathematical Sciences. Senior research scientist in Kazan Branch of Joint Supercomputer Center, Scientific Research Institute of System Analysis, Russian Academy of Sciences; senior research scientist in Higher Institute of Information Technology and Intelligent Systems of Kazan Federal University.



0000-0002-5794-5326

e-mail: dennis.demidov@gmail.com**Aleksandr Mikhailovich Elizarov**

Doctor of Physics and Mathematics, Professor of Kazan Federal University, Merited Scientist of the Tatarstan Republic, Director of Kazan Branch of Joint Supercomputer Center, Scientific Research Institute of System Analysis, Russian Academy of Sciences. Member of the American Mathematical Society (AMS), German Association of Mathematicians and Mechanics (GAMM) and the International Society for Industrial and Applied Mathematics (SIAM). Author of more than 300 publications, (including 12 monographs).



0000-0003-2546-6897

e-mail: amelizarov@gmail.com