Large-scale analysis of energy system vulnerability using in-memory data grid

A V Edelev[1], I A Sidorov[2], S A Gorsky[2] and A G Feoktistov[2]

[1]Melentiev Energy Systems Institute, Lermontov St., 130, Irkutsk, Russia, 664033
[2]Matrosov Institute for System Dynamics and Control Theory SB RAS, Lermontov St., 134, Irkutsk, Russia, 664033

E-mail: flower@isem.irk.ru

Abstract. Nowadays, determining critical components of energy systems is a relevant problem. The complexity of its solving increases significantly when it is necessary to take into account the simultaneous failures of such components. Usually, in problem-solving, processing a large number of failure variants and their consequences is required. Processing such data using traditional relational database management systems does not allow us to quickly identify the most critical components. In the paper, our successful practical experience in applying an in-memory data grid within large-scale analyzing of the energy system vulnerability is provided. The experimental analysis showed the good scalability of distributed computing and significant reduction in data processing time compared to using an open-source SQL relational database management system. In developing and applying the distributed applied software package for solving the aforementioned problem we have used the Orlando Tools framework. Within its applying, we have implemented continuous integration of the package software taking into account the preparing and processing of subject-oriented data through the in-memory data grid.

1. Introduction

Nowadays, high-performance computing technologies, including software and hardware infrastructures for distributed computing, continue to develop. Their capabilities are expanding. Therefore, these technologies are becoming much more complex. Developers and end-users of scientific applications based on the workflow use face challenges in effective preparing, processing, and analysing subject-oriented data in the computing process. To this end, we discuss applying a modern technology of the distributed data storage in relation to large-scale analysis of the energy system vulnerability.

The resilience is the ability of a system to prevent damage before disturbances, mitigate losses during these events, and improve the recovery capability after eliminating their consequences [1]. A disturbance represents a general outage event like natural disasters or man-made disruptions that can cause high impact on a system.

Vulnerability and recoverability are two properties of resilience. The vulnerability reflects the scale of negative consequences that are owing to the disturbance impact on a system. The recoverability characterizes the rate of the system recovery after a disturbance.

Processes associated with the implementation of the aforementioned properties under disturbances are demonstrated in Fig. 1. Here, we have specialized the general scheme of the disturbance impact on the system represented in [2]. Within our new scheme, we consider simultaneous failures of system

components resulting from several disturbances.

A system operates in the stable original state until a disturbance that occurs impacts on a system at the time t_e. The extension of a disturbance and its consequences reach their maximum at the time t_d. The system recovery begins at the time t_b. The recovered state of a system is achieved at the time t_f and supported afterwards. The function $f(t)$ shows the system performance measured before, during, and after a disturbance.

In the paper, we focus on the vulnerability property of resilience. The analysis of the energy system vulnerability addresses the following problems [3]:

- Evaluation of the consequences of a disturbance impact on the energy consumers,
- Determination of the most vulnerable (critical) components.

An energy system component can be considered critical if the failure of that component due to the disturbance impact causes the large system performance degradation [3]. The system response to the disturbance impacts depends on the combination of many factors. They include the external and internal conditions of the system operation, disturbance characteristics (type, location, damage, its duration, etc.), acceptable measures to enhance the system resilience, and various criteria of sharing these factors. In addition, there are many scenarios of the system responses to disturbances and mitigating their consequences.

Traditionally, energy systems are described in the form of a direct graph or network, where nodes represent different components, and arcs show the physical connections among them [3]. Network-based approaches can be divided into topology-based and flow-based methods. The topology-based methods model the energy systems only based on their topologies. The flow-based methods are preferable because they provide more realistic modeling physical processes in the energy systems. In applying flow-based methods, the graph parameters (changing weights of arcs, increasing length of paths between nodes, etc.) are taken into account in evaluating the criticality of energy system components in the case of their failures. In each case, their combination and boundary values are defined by the problem specifics.

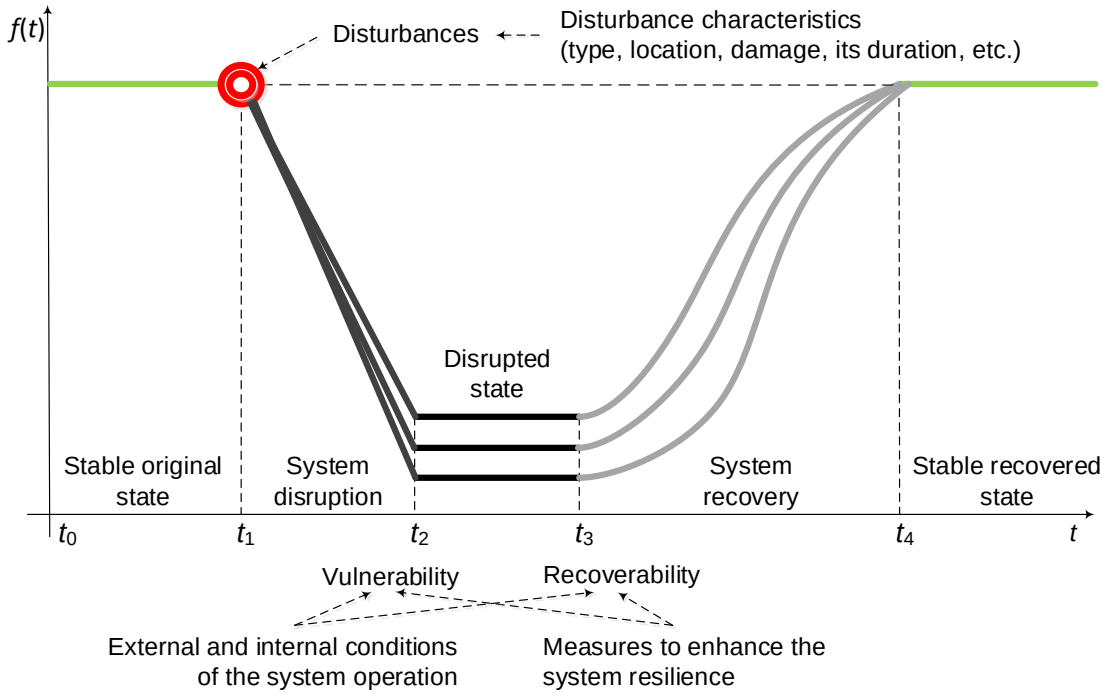

Figure 1. The system performance under a disturbance.

The vulnerability analysis provides the study of system response space. Obviously, such an analysis has a combinatorial character. It is characterized by high computational complexity and intensive operation with the database.

In practice, the study of various variants of simultaneous component failures for an acceptable time requires the use of high-performance computing systems. Even when using high-performance computing, studying the simultaneous failure combinations of more than four components is difficult.

Moreover, for both problems of the vulnerability analysis, the significant problem-solving time is largely owing to the processing model data using traditional relational database management systems.

In the paper, we propose a new approach based on applying an in-memory data grid within large-scale analysis of the energy system vulnerability. Our successful practical experience confirms the advantages of the proposed approach. Within the approach, problems can be solved using various computing resources (personal computer, high-performance computing cluster, etc.), depending on their dimension.

We apply the Orlando Tools framework for the development of scientific applications (distributed applied software packages) [4]. It provides parallel and distributed computing in both the homogenous and heterogeneous distributed environments. In the application development process, Orlando Tools supports the modification and continuous integration of applied and system software taking into account different characteristics of computational resources.

Orlando Tools provide a more large spectrum of capabilities for continuous integration related to creating and using packages in comparison with the well-known tools [5-7]. The fundamental basis of their functioning is a new conceptual model of the computing environment. This model supports the specification, planning, and execution of software continuous integration processes taking into account the subject-oriented data and specifics of solved problems.

The paper is structured as follows. Next section provides a shot review of related works. The problem formulation is considered in Section 3. Section 4 describes the distributed applied software package for analyzing the energy system vulnerability. The experimental analysis of the energy system vulnerability is reported in Section 5. Finally, Section 6 presents the conclusions.

2. Related work

Solving the aforementioned problems related to studies of the energy system vulnerability involves the need for big data processing within workflow (problem-solving scheme) executing [8].

As a rule, a public access computer center provides users with a centralized data storage system with a relational data scheme. So, for example, in the Irkutsk supercomputer center, the Firebird database server is used.

It is an open-source SQL relational database management system [9]. Firebird is very popular since it runs on various platforms. Among them are Linux, Microsoft Windows, macOS, etc.

At the same time, transferring processes of data storage and processing from hard drives to RAM allows us to significantly reduce the time of data exchange in executed applications [10]. This principle of data storage and processing is implemented through an In-Memory Data Grid (IMDG).

IMDG is a distributed data storage that is completely in RAM. It is similar to the multi-threaded hash-table, where data elements are stored by keys. IMDGs are developed to support high data operation scalability through data distribution between several nodes of a distributed computing environment. Unlike traditional storage systems, we can use any data element as a key or value in IMDG.

Such known IMDG systems are Hazelcast, Infinispan, Ehcache, ScaleOut StateServer, Red Hat JBoss Data Grid, Ncache, GridGain Enterprise Edition, Oracle Coherence, IBM WebSphere Application Server, and Terracotta Enterprise Suite [11]. At that, GridGain Enterprise Edition and Hazelcast are actively used in the electric power industry to study simultaneous failures in real time [12].

GridGain Enterprise Edition is based on the free Apache Ignite system [13]. Unlike other similar systems, Apache Ignite has a convenient and compact implementation of its application interface in C++ and supports the SQL use. This is very important at the stages of processing and analysis of computation results.

We have applied Apache Ignite for data processing in our study related to a large-scale analysis of the energy system vulnerability. In contrast with GridGain, Orlando Tools provide end-users of developed packages with the free automated support for the delivery and deployment of Apache Ignite on dedicated resources.

3. Problem formulation

Usually an energy system is represented as a network $G=(X,E)$, where X is a set of $n>0$ nodes, $E \in \left\{(i,j): i,j \in \overline{1,n}, i \neq j\right\}$ is a set consisting of $m>0$ arcs. Each arc $(i,j) \in E$ represents energy resource transportation, i and j are the starting and ending nodes of the arc (i,j), respectively. The flow over the arc $(i,j) \in E$ and its capacity are denoted by x_{ij} and b_{ij}, respectively.

The real energy system like the natural gas or power supply system can contain many producers and consumers. Let us add a common source connecting to each producer and common sink joined by each consumer for the graph G. Then the multi-source and multi-sink problem of energy distribution over the system network can be reduced to the search of the maximum flow w_{st} with the lowest cost between the common source and sink. Such an approach is often used in energy system studies (see, for example, [14]). Wherein, the capacities of artificial arcs connected with the common source and sink are equal to production and demand values, respectively. Thus, this problem can be formulated as follows:

$$c_{ij}x_{ij} \to \min, \tag{1}$$

$$\sum_{i \in N_j^+} x_{ij} - \sum_{i \in N_j^-} x_{ji} = \begin{cases} -w_{st}, \text{ if } j=s, \\ w_{st}, \text{ if } j=t, \\ 0 \text{ otherwise}, \end{cases} \tag{2}$$

$$0 \le x_{ij} \le b_{ij}, \tag{3}$$

where c_{ij} is the flow cost over $(i,j) \in E$, N_j^+ is the subset of input arcs for node j, N_j^- is the subset of output arcs of node j. Equation (2) ensures that the input flow and output flow for any node will be equal. The flow constraints in equation (3) ensure that the flow over any arc will be non-negative and will not exceed its capacity.

The problem (1)-(3) is the simplest energy system model to evaluate the system performance degradation under a disturbance impact. It underlies a task of identifying and ranking critical components and sets of components.

Our approach to identify and rank the energy system critical components is based on the failure sets generation [15] and usage of the Monte Carlo simulations to study the energy system behavior [14]. As defined in [15], a failure set is a specific combination of the energy system nodes and arcs that fail simultaneously and characterized by its size or the number of failed components. The criticality of a component or a set of components is defined as the vulnerability of the system to failure in a specific component, or set of components.

The size k of failure set should be selected by the researcher depending on the total number of network elements that is equal $n+m$. In practice, the following values of $n+m$ and k are relevant: $n+m \ge 100$ and $1 \le k \le 5$. For these reasons, k should not exceed 3 or 4 since the number of possible failure sets is equal to $(n+m)!/((n+m-k)\times k!)$.

Obviously, that the number of possible failure sets grows rapidly with increasing k. This is one of the main problems of the storing and processing model data for the traditional relational database management systems. Often, such systems cannot cope with the data flow and become a performance bottleneck in processing computation results.

We simulate the energy system operation with different failure sets and evaluate the consequences

of their impact on the energy consumers. Each Monte Carlo simulation consists in generating a large number of samples of system response on the components failure and counting those where the total energy resource shortage exceeds the specified threshold [16]. The higher is a share of such samples the more critical is a failure set [17]. As a result, we get also a list of the more critical components.

4. Scientific application for analyzing the energy system vulnerability

We have developed a scientific application (distributed applied software package) for analyzing the energy system vulnerability using the Orlando Tools framework. Unlike other tools for developing scientific applications, Orlando Tools supports the intensive evolution of algorithmic knowledge, adaptation of existed and designing new ones. It automates the non-trivial technological sequence of the collaborative development and use of packages including the continuous integration, delivery, deployment, and execution of package modules in a heterogeneous distributed environment [18-21].

Aspects of the energy system vulnerability analysis package development are considered in detail in [22]. The subject domain and all parameters of the package including problem-solving algorithm are described in [23]. The package description is stored on the Orlando Tools server.

Figure 2 shows the problem-solving scheme (workflow) s_1 in the package. Using this scheme, we find the energy system critical components applying Apache Ignite data grid to process Monte Carlo simulation data. The operation *IgniteStart* deploys Apache Ignite first in the scheme s_1 to orginise the data grid.

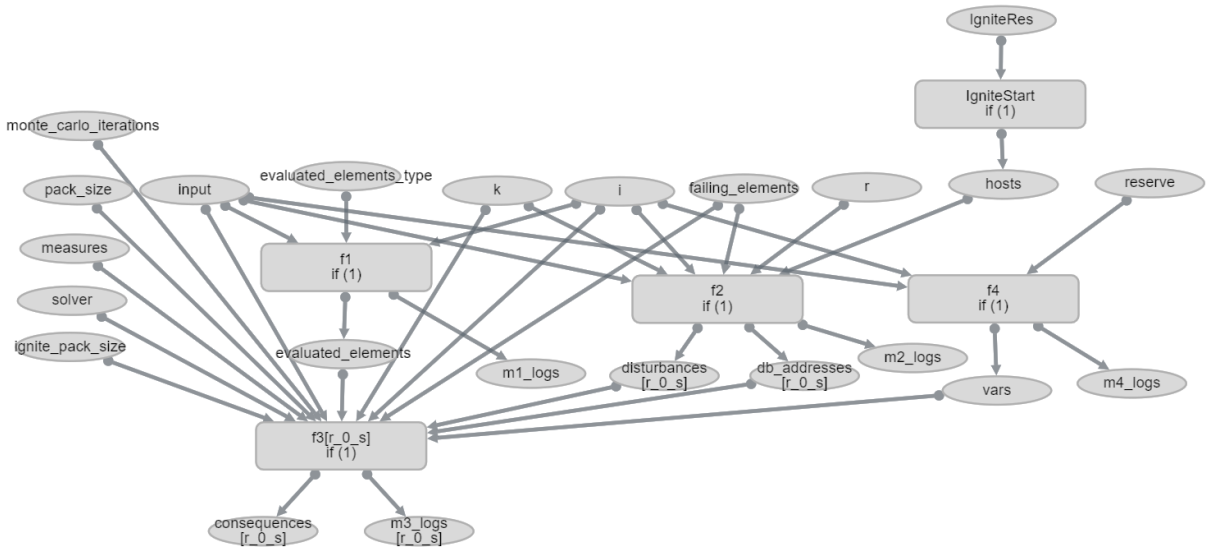

Figure 2. Problem-solving scheme based on the Apache Ignite use.

The scheme s_1 enables the synchronous evaluation of failure sets of the size k. For this purpose a set of failure sets of the specified size k generated by the operation f_2 of the scheme s_1 is divided into r subsets. The operation f_2 use makes it possible to form different job flows for various k. These flows differ in the computational load for their perform. Thus, they can be performed on heterogeneous resources with different computational characteristics. This increases the efficiency of their use.

Consequences of the energy system nodes and arcs failure are evaluated during parallel Monte Carlo simulation by r instances of the operation f_3. Negative consequences, for example, unsatisfied demands are calculated for a list of consumers created by operation f_1. The distribution of data sets by instances is shown in Figure 3.

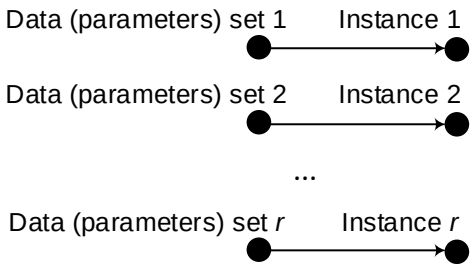

Figure 3. Data distribution.

The operation f_4 determines the size of the one sample output data sent to the Apache Ignite data grid.

The operations f_1, f_2, f_3, f_4, and *IgniteStart* are implemented by the modules m_1, m_2, m_3, m_4, and m_5, respectively. The modules m_2, m_3 and m_5 operate with Apache Ignite.

Figure 4 shows the problem-solving scheme s_2. Using s_2, we find the critical components applying Firebird for data processing. The operations $f_1 - f_4$ perform the same actions as in the scheme s_1.

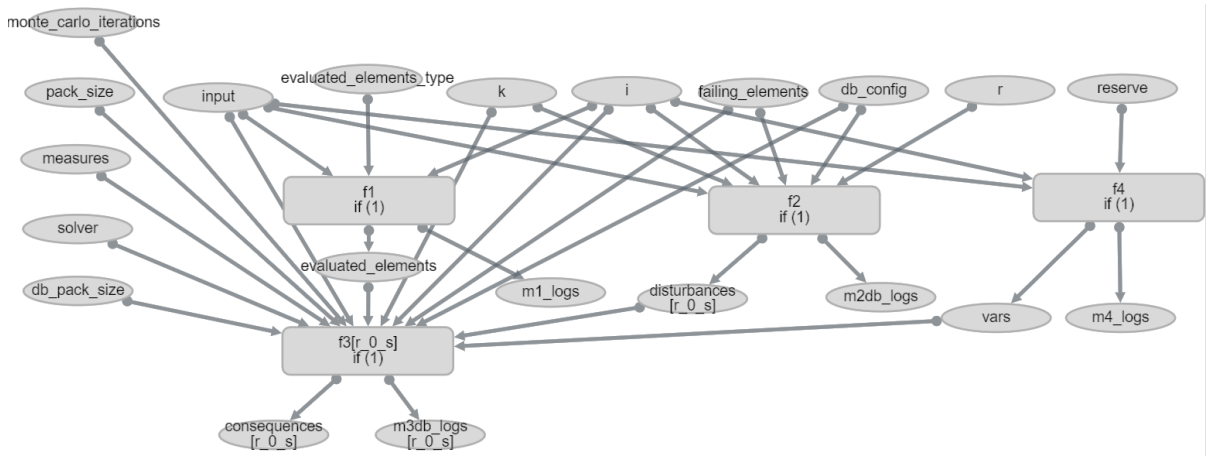

Figure 4. Problem-solving scheme based on the Firebird use.

The operations f_1 and f_4 are implemented by the same modules m_1 and m_4. However, the operations f_2 and f_3 are implemented by the new modules m_6 and m_7. These modules operate with Firebird.

Thus, both schemes are similar. The differences are in the parameters for operating with different databases and implementation of the operations f_2 and f_3. In addition, scheme s_1 includes the additional operation *IgniteStart*.

The resource configurations, information about installing and testing modules on these resources, and Apache Ignite configuration are stored on the Orlando Tools server.

5. Experimental analysis

Within the experiment, we determine critical components of the Unified Gas Supply System of Russia. The system network consists of the 382 nodes, including 28 natural gas sources, 64 natural gas consumers, 24 underground natural gas storages, and 266 compressor nodes. In addition, it includes 486 arcs representing main natural gas pipelines and branches to distribution networks.

Within this study, failure sets of size 1 were created from selected 415 arcs and 291 nodes (natural gas sources, underground storages, and compressor stations). The components selection was carried out by experts taking into account their practical experience in solving similar problems.

In the process of computing, we used nodes of the high-performance computer (HPC) cluster of the public access Irkutsk Supercomputer center as the distributed computing environment [24]. They have the following characteristics: 2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 16 MB L3 cache, 4 FLOP/cycle, 64 GB RAM DDR3-1600.

During data processing, we compared the use of the Firebird 2.5 SuperServer and Apache Ignite 2.8 data grid. The database running under Firebird was configured to use asynchronous data writes for improving Firebird's performance during large batch operations. It means that the modified or new records are put into the memory cache for periodic flushing to hard disk by the operating system [9].

Apache Ignite is running in the partitioned mode that is the most scalable distributed cache mode. In this mode, the overall data is divided equally into partitions. Then all partitions are split equally between participating nodes organized from an Apache Ignite. This approach allows us storing as much data as can be fit in the total memory available across the Apache Ignite nodes of the distributed computing environment.

The location of the main computing components at using Apache Ignite and Firebird is shown in Figure 5(a) and Figure 5(b), respectively. The aforementioned HPC-cluster was used in the experiment. Firebird 2.5 SuperServer and Apache Ignite 2.8 were installed on the dedicated node node001. On the same node, the main process was launched. Additional processes were connected to the main process. The Apache Ignite processes were launched through the task queue on compute nodes through the local PBS Torque resource manager. The modules of the package are executable programs for OS Linux. They are located in the user folder of the HPC-cluster. The order of launching package modules is set by the above-mentioned schemes. During computing, package modules are launched by the Orlando Tools computation manager through the cluster task queue using the SSH protocol.

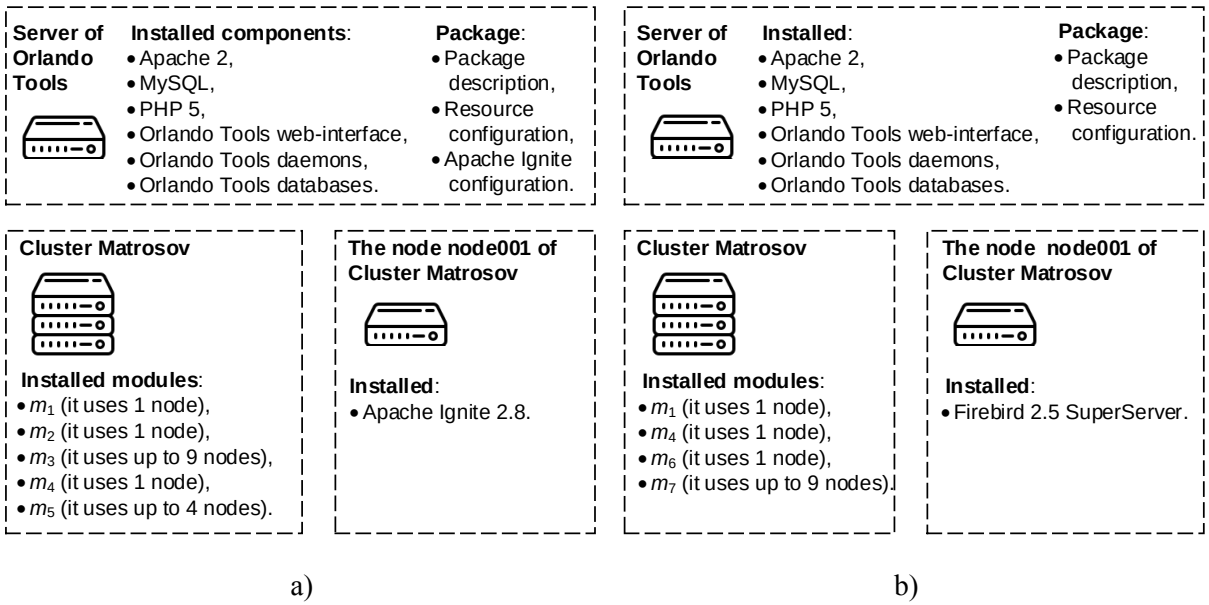

Figure 5. Location of computational components with applying Apache Ignite (a) and Firebird (b).

Figure 6 shows the problem-solving time relative to the different number of iterations of the Monte Carlo method per disturbance applying the Firebird and Apache Ignite. During computing, 9 nodes of the high-performance computer cluster were used. Apache Ignite used 1 node. Applying the Apache Ignite provides a significant decrease in the problem-solving time in comparison with Firebird. This is

due to the fact that the time spent on data writing to the database when using Firebird reaches 72%. In the case of the Apache Ignite, this share does not exceed 6%.

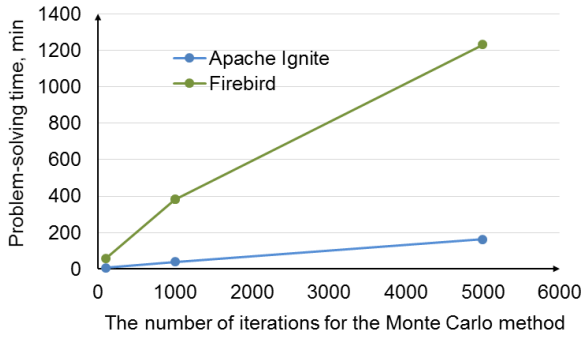

Figure 6. Problem-solving time vs. the number of iterations for the Monte Carlo method.

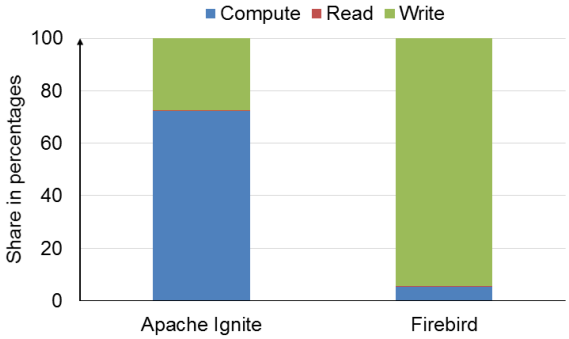

Figure 7. Share of operations Compute, Read, and Write in the total problem-solving time.

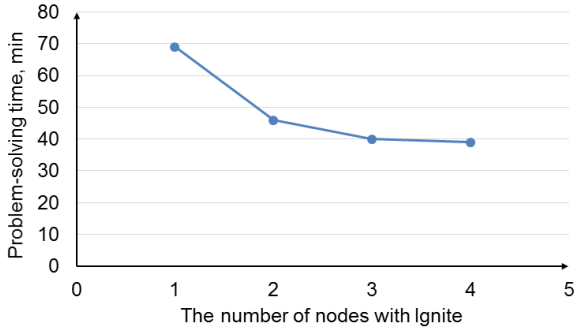

Figure 7. Data processing scalability in problem-solving.

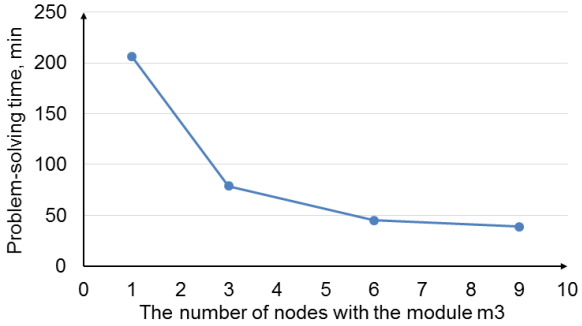

Figure 8. Computing scalability in problem-solving.

Share of operations Compute, Read, and Write in the total execution time of the operation f_3 is demonstrated in Figure 7.

We provide two additional experiments. Figure 8 shows data processing scalability. We can see that the problem-solving time is decreased with the increase of the number of the Apache Ignite nodes intended to process model data. The computing scalability is also validated. Finally, Figure 9 demonstrates decreasing the problem-solving time with increasing of the number of nodes, in which the module m_3 of the scheme s_1 runs in parallel.

6. Conclusions

IMDG is a novel data storing technology for distributed applied software packages. It provides higher performance and scalability in comparison with the traditional rational databases. The advantages of IMDG we demonstrate in the practical study. In the paper, we address the relevant problem of determining the critical components of the energy system. A failure of one or more such critical components can lead to serious damage in the social, economic or political spheres of human activity, and threaten the national security. To determine the critical components, we apply a method of combinatorial modeling that can model simultaneous failure of two and more components. This method is characterized by high computational complexity. To this end, we use Apache Ignite data grid in processing model data in the distributed computing environment.

The experimental analysis obviously demonstrates a significant decrease in the problem-solving time through the distributed data processing with Apache Ignite. The results of this analysis show the substantial advantages of using Apache Ignite compared to traditional relational database management systems. In addition, we demonstrate the scalability of data processing and computing in problem-

solving scheme executing. We have used the Orlando Tools in developing and applying the distributed applied software package for solving the problem of determining the critical components of the energy systems. In particular, we have applied Orlando Tools for continuous integration within the framework of the package development using new technology in preparing and processing of subject-oriented data.

In the future, we suppose to use the capabilities of Apache Ignite for the intellectual analysis of intermediate results of computations in order to identify correlations between criteria of the importance for the critical components.

7. Acknowledgments

The study was supported by the basic research program of SB RAS, project no. III.17.5.1. In addition, the development of tools for developing and applying scalable scientific applications (distributed applied software packages) was provided within the same program, project no. IV.38.1.1.

References

[1] Zio E 2016 Challenges in the vulnerability and risk analysis of critical infrastructures *Reliab. Eng. Syst. Safe.* **152** 137–150

[2] Almoghathawi Y, Barker K and Albert L A 2019 Resilience-driven restoration model for interdependent infrastructure networks *Reliab. Eng. Syst. Safe.* **185** 12–23

[3] Johansson J and Hassel H 2012 Modelling, simulation and vulnerability analysis of interdependent technical infrastructures *Springer Series in Reliability Engineering. Risk and Interdependencies in Critical Infrastructures: A Guideline for Analysis* ed P Hokstad, I B Utne and J Vatn (London: Springer-Verlag) pp 49–66

[4] Feoktistov A, Gorsky S, Sidorov I, Bychkov I, Tchernykh A and Edelev A 2020 Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment *Comm. Com. Inf. Sc.* **1087** 18–32

[5] Gruver G 2016 *Start and Scaling Devops in the Enterprise* (BookBaby) p 100

[6] Shahin M, Babar M A and Zhu L 2017 Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices *IEEE Access* **5** 3909–3943

[7] Wolff E 2017 *A Practical Guide to Continuous Delivery* (Addison-Wesley) p 265

[8] Fuchs M, Teichmann J, Lauster M, Remmen P, Streblow R and Muller D 2017 Workflow automation for combined modeling of buildings and district energy systems *Energy* **117** 478–484

[9] from https://firebirdsql.org/

[10] Zhang H, Chen G, Ooi B C, Tan K and Zhang M 2015 In-memory big data management and processing: A survey *IEEE Trans. Knowledge and Data Eng.* **27(7)** 1920–1948

[11] https://www.predictiveanalyticstoday.com/top-memory-data-grid-applications

[12] Zhou M and Yan J 2018 A new solution architecture for online power system analysis *CSEE J. Power Energy Systems* **4(2)** 250–256

[13] Bhuiyan S, Zheludkov M and Isachenko T 2017 *High Performance in-memory computing with Apache Ignite* (Morrisville: Lulu.com) p 352

[14] Praks P and Kopustinskas V 2016 Identification and ranking of important elements in a gas transmission network by using ProGasNet *Proc. of the European Safety and Reliability Conf. on Risk, Reliability and Safety: Innovating Theory and Practice* ed L Walls, M Revie and T Bedford (CRC Press) pp 1573–1579

[15] Jonsson H, Johansson J and Johansson H 2008 Identifying critical components in technical infrastructure networks *P. I. Mech. Eng. O.-J. Ris.* **222(2)** 235–243

[16] Zio E 2018 The future of risk assessment *Reliab. Eng. Syst. Safe.* **177** 176–190

[17] Robert C and Casella G 2004 *Monte Carlo statistical methods* (New York: Springer-Verlag) p 649

[18] Bychkov I V, Oparin G A, Tchernykh A, Feoktistov A G, Gorsky S A and Rivera-Rodriguez R 2018 Scalable Application for the Search of Global Minima of Multiextremal Functions *Optoelectron. Instrum. Data Process.* **54(1)** 83–89

[19] Feoktistov A, Gorsky S, Sidorov I, Kostromin R, Edelev A and Massel L 2019 Orlando Tools: Energy Research Application Development through Convergence of Grid and Cloud Computing. *Commun. Comput. Inf. Sci.* **965** 289–300
[20] Bychkov I, Oparin G, Feoktistov A, Sidorov I, Gorsky S, Kostromin R and Edelev E 2019 Subject-oriented computing environment for solving large-scale problems of energy security research *J. Phys. Conf. Ser.* **1368** 052030-1–052030-12
[21] Tchernykh A, Feoktistov A, Gorsky S, Sidorov I, Kostromin R, Bychkov I, Basharina O, Alexandrov A and Rivera-Rodriguez R 2019 Orlando Tools: Development, Training, and Use of Scalable Applications in Heterogeneous Distributed Computing Environments *Commun. Comput. Inf. Sci.* **979** 265–279
[22] Feoktistov A, Gorsky S, Sidorov I, Bychkov I, Tchernykh A and Edelev A 2020 Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment *Commun. Comput. Inf. Sci.* **1087** 18-32
[23] Edelev A, Beresneva N, Gorsky S, Sidorov I and Feoktistov A 2019 Representation of Subject Knowledge from the Field of Vulnerability Analysis of Energy Systems in Distributed Applied Software Packages *Advances in Intelligent Systems Research* **169** 184–188
[24] http://hpc.icc.ru/