

An Approach to Implementing High-Performance Computing for Problem Solving in Workflow-based Energy Infrastructure Resilience Studies

[Alexander Feoktistov](#)^{*}, Alexei Edelev, [Andrei Tchernykh](#), [Sergey Gorsky](#), Olga Basharina, Evgeniy Fereferov

Posted Date: 2 November 2023

doi: 10.20944/preprints202311.0065.v1

Keywords: energy systems; resilience; vulnerability; high-performance computing; IMDG; scientific workflows



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

An Approach to Implementing High-Performance Computing for Problem Solving in Workflow-Based Energy Infrastructure Resilience Studies

Alexander Feoktistov ^{1,*}, Alexei Edelev ², Andrei Tchernykh ^{3,4}, Sergey Gorsky ¹,
Olga Basharina ^{1,5} and Evgeniy Fereferov ¹

¹ Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences, 664033 Irkutsk, Russia. agf@icc.ru (A.F.); gorsky@icc.ru (S.G.); ouyb@icc.ru (O.B.); fereferov@icc.ru (E.F.)

² Melentiev Energy Systems Institute of the Siberian Branch of the Russian Academy of Sciences, 664033 Irkutsk, Russia. flower@isem.irk.ru (A.E.)

³ CICESE Research Center, 22860 Ensenada, Mexico. chernykh@cicese.mx (A.T.)

⁴ Ivannikov Institute for System Programming of the Russian Academy of Sciences, 109004, Moscow, Russia.

⁵ Ural State University of Economics, 620144 Yekaterinburg, Russia.

* Correspondence: agf@icc.ru

Abstract: Implementing High-Performance Computing to solve In-Memory Data Grid (IMDG)-based energy infrastructure resilience research problems in a heterogeneous environment presents a challenge to workflow management systems. Large-scale energy infrastructure needs multi-variant planning and tools to allocate and dispatch resources in IMDG taking into account the subject domain specificity, resource characteristics, and constraints and quotas for resource use. To that end, we propose a workflow management system using our Orlando Tools framework. To scale computing resources, we provide their integration and use corresponding software to determine key application parameters that can significantly impact the processed data size and the required number of allocated resources. We automated the IMDG cluster launch for the workflow executions. To demonstrate the advantage of our solution, we apply it to evaluate the resilience of the existing energy infrastructure model. Compared to similar approaches, our approach explores large infrastructures by modeling many simultaneously failed elements of different types up to the number of network elements. In terms of problem-solving efficiency and resource use, we achieve near-linear speedup with increasing the number of nodes of each resource.

Keywords: energy systems; resilience; vulnerability; high-performance computing; IMDG; scientific workflows

1. Introduction

Nowadays, the economy and the welfare of society depend significantly on the resilience of critical infrastructures with various disturbances [1]. An energy infrastructure is one of the vital infrastructures because it ensures the functioning of other dependent infrastructures. Therefore, considering the large scale of this integrated and multi-level infrastructure, the study of its resilience is undoubtedly a challenge. Many experts in the field pay special attention to it [2,3]. In particular, the following key characteristics of energy infrastructure resilience are identified qualitatively: anticipation, resistance, adaptation, and recovery. The essence of anticipation is predicting disturbance events and taking preventive actions. Resistance is defined as the ability of an energy infrastructure to absorb the impact of disturbance and mitigate its consequences with minimal effort. The adaptation reflects how an energy infrastructure can self-organize to prevent system performance degradation and overcome disturbances without restoration activities. Finally, the recovery shows the ability to restore the system functioning to its original operating level.

To assess the resilience of energy infrastructures, it is necessary to investigate the changes between the states of the energy infrastructure before, during, and after the disturbance.

In general, such changes are represented by resilience curves (see [4]). These curves show the performance evolutions over time under a disturbance scenario for different energy infrastructure configurations (Figure 1). Each curve reflects the following states:

- Pre-disturbance state ($t < t_j$) that shows the stable performance $P(t)$,
- Disturbance progress state ($t \in [t_j, t_{j+1}]$) demonstrating the occurrence of a disturbance and a decrease in the performance $P(t)$ eventually,
- Post-disturbance degraded state ($t \in [t_{j+1}, t_{j+2}]$) characterized by a rapid drop in the performance $P(t)$ from t_{j+1} ,
- Restoration state ($t \in [t_{j+2}, t_{j+3}]$) when the performance $P(t)$ gradually increases from t_{j+2} to t_{j+3} , and the energy infrastructure reaches a new stable condition,
- Post-restoration state ($t \in [t_{j+3}, t_{j+4}]$), in which the performance $P(t)$ reaches a level comparable to that before the disturbance.

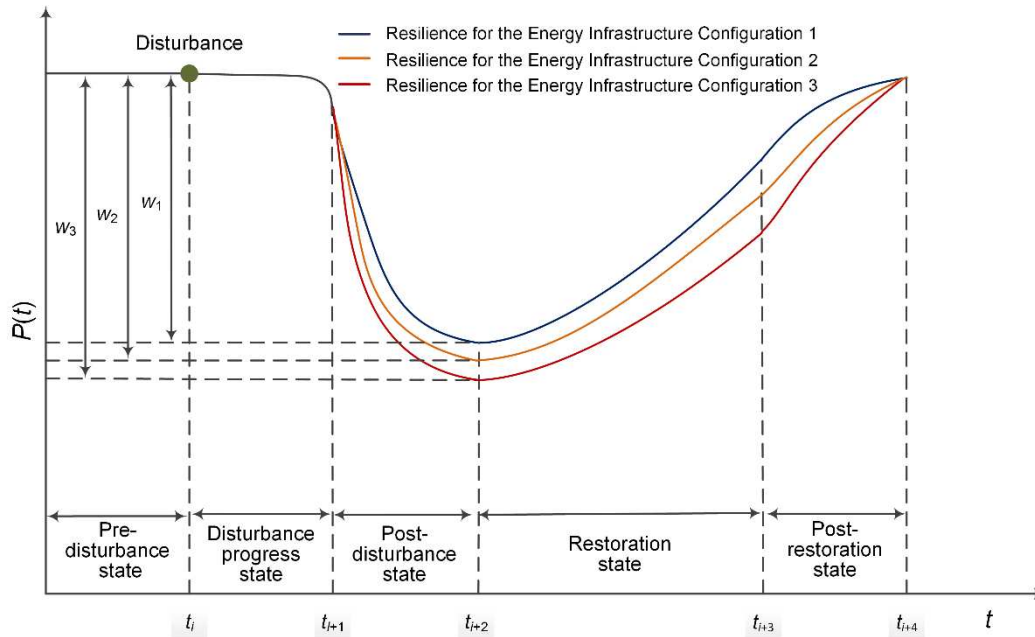


Figure 1. Resilience curves.

Puline et al. [4] introduce metrics to quantify resilience based on the shape and size of curve segments. Let a disturbance scenario contains n_p periods and the vector $s^\tau = (s_1^\tau, s_2^\tau, \dots, s_m^\tau)$ describes the state of the energy infrastructure at the period $\tau \in \overline{1, n_p}$. Next, the performance measure $y(S, \tau): s^\tau \rightarrow \mathbf{R}$ maps the energy infrastructure state s^τ from the set $S = \{s^1, s^2, \dots, s^{n_p}\}$ to a scalar value. The summary metric $w(S, [a, b]): (y(S, i_1), y(S, i_2), \dots, y(S, i_k)) \rightarrow \mathbf{R}$ then maps a curve segment to a scalar value, $a = i_1$, $b = i_k$, $i_1 < i_2 < \dots < i_k$, $i_1, i_2, \dots, i_k \in \overline{1, n_p}$. The following data sources can be used to plot resilience curves: retrospective information, natural experiment results, simulation or analytical modeling, artificial neural networks, network flow methods, etc.

Within our study, we consider one of the fundamental metrics of energy infrastructure resilience called vulnerability. A vulnerability is studied within a Global Vulnerability Analysis (GVA) [5]. The purpose of such an analysis is to evaluate the extent to which the performance degradation of the energy infrastructure configuration depends on the disturbance parameters [6]. GVA is based on modeling a series of disturbance scenarios with a large number of failed elements. An energy infrastructure configuration includes the following entities: energy facility descriptions, as well as data about the natural-climatic, technological, social, and economic conditions for the operation of these facilities. Analyzing the dependence of the performance degradation because of the disturbance parameters is a straightforward way to compare different energy infrastructure configurations in terms of vulnerability. The steeper and deeper the slope of the resilience curve segment in the resistance and absorption states for a specific configuration, the more vulnerable this configuration is to disturbance parameters. For example, the curve segment from t_i to t_{i+2} in Figure 1 reflects the

resistance state in the form of a sharp degradation in the energy infrastructure performance $P(t)$. The metrics $w_1(S, [t_i, t_{i+2}])$, $w_2(S, [t_i, t_{i+2}])$, and $w_3(S, [t_i, t_{i+2}])$ equal $P(t_i) - P(t_{i+2})$ the vulnerability of the energy infrastructure due to the performance degradation in the two states above. In this example, $w_1 < w_2 < w_3$. Therefore, Energy Infrastructure Configuration 1 is the least vulnerable.

Researchers involved in implementing GVA are faced with its high computational complexity. It is necessary to create sets of possible disturbance scenarios by varying the values of their parameters for each energy infrastructure configuration. However, the problem-solving process is well parallelized into many independent subtasks. Each subtask investigates the vulnerability of a particular energy infrastructure configuration to a set of disturbance scenarios. In this context, we implement GVA based on the integrated use of HPC, a modular approach to application software development, parameter sweep computations, and IMDG technology.

The rest of the paper is structured as follows. Section 2 briefly reviews approaches for implementing high-performance computing and scientific applications in power system research. Section 3 provides the main aspects of performing GVA. Section 4 presents the workflow-based distributed scientific application that implements the proposed approach. Section 5 discusses the results of computational experiments. Finally, Section 6 concludes the paper.

2. Related Work

In this brief review, we consider the most appropriate approaches to implementing the following three aspects of computing organization that drive our problem-solving:

- HPC for solving similar problems in general,
 - The use of the IMDG technology to increase the efficiency of storing and processing the computation results in comparison with traditional data storage systems,
 - Capabilities of software systems to create and execute the required problem-solving workflows.
- Based on these considerations, we define the aspects of our approach.

2.1. High-performance computing implementations

An analysis of the available literature describing the software and hardware to study power systems allows us to draw the following conclusions. The software developed in this research field generally focuses on solving a specific problem with a predefined dimension [7]. The applied software adapts to the capabilities of the available software and hardware of computing systems [8]. Using a similar approach dramatically simplifies software development. However, the problem specification capabilities and increasing its dimension are usually limited. Moreover, in this case, it is impossible to distinguish a unified way to implement parallel or distributed computing for a wide range of problems. At the same time, solving many tasks requires capabilities of computing environments that are dynamically determined and changed for the problem dimensions and algorithms used.

We distinguish the following three general approaches:

- *Developing multi-threaded programs for systems with shared memory based on the OpenMP or other standards* [8–10]. This approach has high-level tools for developing and debugging parallel programs. However, the structure of the programs should not have a high complexity. As a rule, such an approach allows solving problems of relatively low dimensionality. Otherwise, it is necessary to use special supercomputers with shared memory [8] or implement computations using the capabilities of other approaches.
- *Implementing MPI programs* [9,10]. This approach focuses on homogeneous computing resources and allows us to solve problems of any complexity, depending on the number of resources. A more complex program structure, as well as a low level of tools for software development and debugging, characterize this approach. In general, however, focusing on homogeneous computing resources allows us to solve problems of any complexity, depending on the available computing resources.

- *Creating distributed applications for Grid and Cloud Computing* [11]. There is a large spectrum of tools for implementing distributed computing that provide different levels of software for development, debugging, and usage by both developers and end-users [12]. Undoubtedly, this approach can solve problems of high dimensions and ensure computational scalability. However, within such an approach, the challenge is allocating and balancing resources with different computational characteristics.

To solve the problems related to GVA, we based on the third approach using the capabilities of the first approach to implement parallel computing. In addition, we use the IMDG technology to speed up data processing.

2.2. IMDG Tools

The IMDG-based data storage systems have an undeniable advantage in data processing speed compared to traditional databases [13]. Moreover, a large spectrum of tools to support the IMDG technology on the HPC resources is available [14,15]. Each of these tools is a middleware to support distributed processing of large data sets. Hazelcast [16], Infinispan [17], and Apache Ignite [18] are the most popular freely distributed IMDG middleware for workflow-based and job-flow-based scientific applications with similar functionality and performance. They support client APIs on Java, C++, and Python. End-users can use them to create IMDG clusters with distributed key/value caches. Each can have a processing advantage for a particular scenario and dataset.

The Hazelcast platform is a real-time software platform for dynamic operations on data. It integrates high-performance stream processing with fast data storage [19]. Some Hazelcast components are freely distributed under the Hazelcast Community License Agreement version 1.0.

Infinispan is the IMDG tool that provides flexible deployment options and reliable means to store, manage, and process data [20]. It supports and distributes key/value data on scalable IMDG clusters with high availability and fault tolerance. Infinispan is available under the Apache License 2.0.

Apache Ignite is a full-featured, in-memory decentralized transactional key-value database with a convenient and easy-to-use interface for real-time operation of large-scale data, including asynchronous in-memory computing [21]. It supports a long-term memory architecture with an in-memory and on-disk big data accelerator for the data grid, computation grid, service grid, and streaming grid.

Apache Ignite is currently available as an open-source product under the Apache License 2.0. Unlike Hazelcast, the full functionality of Apache Ignite and Infinispan is free.

Within the SQL grid, Apache Ignite provides the operation with a horizontally scalable and fault-tolerant distributed SQL database that is highly ANSI-99 compliant. At the same time, Hazelcast and Infinispan support similar SQL queries with exceptions.

In addition, to solve our problem, the IMDG middleware deployment on the computational nodes needs to be provisioned dynamically during the workflow execution. This is a key requirement. For Hazelcast and Infinispan, this is a time-consuming process. Typically, the Hazelcast and Infinispan end-users first deploy a cluster and only then run data processing tasks on the nodes of the deployed cluster. It was shown in various IMDG-based power system studies [22–24]. From this point of view, Apache Ignite is undoubtedly preferable.

For Apache Ignite, a technique is developed to determine the number of IMDG cluster nodes depending on the predicted amount of memory required to process the data [25]. A similar technique was developed for Infinispan. However, Apache Ignite additionally considers the memory overheads of data storage and disk utilization. Hazelcast only provides examples that can be extrapolated for specific workloads.

We have successfully tested the above-listed advantages of Apache Ignite in practice in our preliminary studies [26] using the Orlando Tools (OT) framework [27]. Other researchers have also reported good results in energy system studies using Apache Ignite [28].

2.3. Workflow Management Systems

Apache Ignite, like Hazelcast, Infinispan, etc., does not support scheduling flows of data processing tasks. Therefore, additional tools are required for data decomposition, aggregation, preprocessing, task scheduling, resource allocation, and load balancing among the IMDG cluster nodes. If the set of data processing operations in a specific logical sequence correlates with the concept of a scientific workflow, the Workflow Management System (WMS) is best suited for scheduling.

A scientific workflow is an information and computation structure. It represents the business logic of a subject domain in applying subject-oriented data and software (a set of applied modules) for solving problems in this domain. WMS is a software tool that partially automates specifying, managing, and executing a scientific workflow according to its information and computation structure. Often, Directed Acyclic Graph (DAG) is used to specify a scientific workflow. In general, the DAG vertices and edges correspondingly represent the applied software modules and data flows between them.

Nowadays, the following WMSs are well-known, supported, developed, and widely used in practice: Uniform Interface to Computing Resources (UNICORE) [29], Directed Acyclic Graph Manager (DAGMan) [30], Pegasus [31], HyperFlow [32], Workflow-as-a-Service Cloud Platform (WaaS) [33], Galaxy [34], and OT.

UNICORE, Condor DAGMan, Pegasus, and OT are at the forefront of traditional workflow management. They are based on a modular approach to creating and using workflows. In the computing process, the general problem is divided into a set of tasks, generally represented by DAG. UNICORE and OT also provide additional control elements of branches and loops in workflows to support the non-DAG workflow representation.

At the same time, the current direction of modern WMSs is comprehensive support for Web services. The use of Web services greatly expands the computational capabilities of workflow-based applications. Web service-enabled WMSs allow us to flexibly and dynamically integrate various workflows created by different developers through the data and computation control in workflow execution. HyperFlow, WaaS Cloud Platform, Galaxy, and OT are successfully developing in this direction.

For scientific applications focused on solving various classes of problems in the field of environmental monitoring, integration with geographic information systems through specialized Web Processing Services (WPSs) is of particular importance. Representative examples of projects aimed at creating and using such applications are the GeoJModel-Builder (GJMB) [35] and the Business Process Execution Language (BPEL) Designer Project (DP) [36]. GJMB and BPEL DP are related to the class of WMSs. GJMB is a framework for managing and coordinating geospatial sensors, data, and applied software in a workflow-based environment. BPEL DP implements the integration of the geoscience model services and WPSs through the use of BPEL-based workflows.

The key capabilities of the WMSs above-listed, which are particularly important from the point of view of our study, are given in Table 1. The systems under consideration use XML-like or scripting languages for the workflow description. Support for branches, loops, and recursions in a workflow allows us to organize computing more flexibly if necessary. OT provides such support in full. An additional advantage of UNICORE, DAG-Man, Pegasus, HyperFlow, Galaxy, GJMB, and OT is the ability to include system operations in the workflow structure. Such operations include pre- and post-processing of data, monitoring the computing environment, interaction with components of local resource managers, etc. HyperFlow, WaaS, Galaxy, GJMB, BPEL DP, and OT support Web services. Moreover, GJMB, BPEL DP, and OT support WPSs.

Table 1. WMS capabilities for supporting.

WMS	Workflow Description Language	Support for Branch / Loop / Recursion in Workflow	Support for System Operations in Workflow	Web Service Support	WPS Support
UNICORE	XML-like	+ / + / -	+	+	-

DAGMa n	Script-like	- / - / -	+	-	-
Pegasus	XML-like	- / - / -	+	-	-
HyperFl ow	JavaScript, JSON	- / - / -	+	+	-
WaaSC P	XML-like	- / - / -	-	+	-
Galaxy	XML-like	- / + / -	+	+	-
GJMB	XML-like	- / - / -	+	+	+
BPEL DP	BPEL	+ / + / -	-	+	+
OT	XML-like	+ / + / +	+	+	+

Table 2 shows the capabilities of the WMSs to support parallel and distributed computing within the execution of workflows. In Table 2, task level means that the tasks determined by the workflow structure are executed on the independent computing nodes. At the data level, a data set is divided into subsets. Each subset is processed on a separate computing node by all or part of the applied software modules used in a workflow. At the pipeline level, sequential executing applied modules for data processing are performed simultaneously on different subsets of a data set. The Computing Environment column indicates the computing systems for which the considered WMSs are oriented are indicated. Most WMSs do not require additional middleware (see Computing Middleware column).

Table 2. WMS capabilities for supporting parallel and distributed computing and data processing.

WMS	Computing Support Level	Computing Environment	Computin g Middlewa re	IMDG Support	IMDG Middlewa re
UNICO RE	Task, data	Grid	-	-	-
DAGMa n	Task, data	Cluster, Grid, Cloud	HTCondor	-	-
Pegasus	Task, data	Cluster, Grid, Cloud	DAGMan	-	-
HyperFl ow	Task	Cloud	-	-	-
WaaSC P	Task	Grid, Cloud	JClouds API	-	-
Galaxy	Task	Cloud	-	-	-
GJMB	Task	Cluster, Grid, Cloud	-	-	-
BPELD P	Task	Cluster, Grid, Cloud	-	-	-
OT	Task, data, pipeline	Cluster, Grid, Cloud	-	+	Apache Ignite

Compared with other WMSs, OT supports all computing support levels for Cluster, Grid, and Cloud. It does not require additional computing middleware. In addition, only OT provides system software to support the IMDG technology based on the Apache Ignite middleware.

The comparative analysis shows that OT currently has the required functionality compared to other WMSs under consideration. Moreover, the system modules for evaluating the memory needed for data processing, resource allocating for IMDG, and Apache Ignite deploying are new components of OT. These modules are developed to support the creation and use of scientific applications in the environmental monitoring of natural areas, including the Baikal Natural Territory (BNT). OT is used to implement HPC for studying power systems.

3. Models, Algorithms, and Tools

The approach proposed in the paper for the implementation of GVA of the energy infrastructure consists of the following three main stages:

- Testing stage,
- Preliminary computing stage,
- Target computing stage.

In the first stage, the modules and workflows of the distributed scientific application for modeling the energy infrastructure vulnerability to external disturbance are tested. The testing process elicits a set of key application parameters that can significantly affect the size of the processed data.

The purpose of preliminary computing is to determine the minimum required size of a set of disturbance scenarios for the GVA implementation. The minimum size of the set of disturbance scenarios is selected based on the required level of reliability in the GVA results. The selected size is used for all energy infrastructure configurations.

Finally, the stage of target computing is aimed at determining the extent of vulnerability for the energy infrastructure.

In the second and third stages, fast data processing is required to better understand the experimental data. Such understanding allows specialists in GVA to dynamically adapt a problem-solving process through changing conditions and parameters of their experiments. Rapid data processing is implemented using the IMDG technology to minimize data movement and disk usage to increase overall computing efficiency. Determining the required number of nodes in the computing environment for data processing in IMDG is based on analyzing the parameters identified in the first stage.

3.1. Model of Evaluating the Data Processing Makespan and Required Memory Size

In the first stage of the proposed approach, we test modules and workflows of the application on environment resources using the OT software manager within continuous integration. Aspects of continuous integration in OT are considered in detail in [37]. The purpose of the testing is to determine the dependency between key parameters of the GVA problem-solving with respect to the data processing on the Apache Ignite cluster.

We evaluate the minimum number of nodes required for the Apache Ignite cluster using the general technique presented in [25]. However, in some cases, the evaluations based on this technique were not accurate. The predicted memory size was insufficient for our experiments with real In-Memory computing [38]. It may be difficult to determine some of the parameters used within this technique. Therefore, we modified the technique to compensate for the memory shortage. In addition, we took into account that disk space is not used in our problem-solving. Thus, we determine the minimum number of nodes as follows:

$$\eta_i^{min} = \left\lceil \frac{d_i^r}{d_i^n(1-k_i^{os})} \right\rceil, \quad (1)$$

$$d_i^r = (1 + n_i^c)(1 + k_i^{ind} + k_i^{bin})k_i^{use}d_i^0 - f_i(n_i^s) + d_i^e, \quad (2)$$

$$d_i^0 = n_i^s d_i^s, \quad (3)$$

where

- $i = \overline{1, n_r}$ is an index of i th resource,
- n_r is a number of resources,
- η_i^{min} is a predicted minimum number of nodes required for the Apache Ignite cluster,
- d_i^n is an available memory size per node in GB,
- d_i^0 is an initial data size in GB,
- d_i^r is an evaluation of the required memory size in GB for data processing as a whole,
- d_i^s is the memory size required for one scenario in GB,

- n_i^c is a number of data backup copies,
- n_i^s is a number of disturbance scenarios,
- $0 < k_i^{os} \leq 1$ is a coefficient of the memory utilization by OS,
- $0 < k_i^{ind} \leq 1$ is a coefficient of the memory utilization for data indexing,
- $0 \leq k_i^{bin} \leq 1$ is a coefficient of the memory utilization for the binary data use,
- $0 < k_i^{use} \leq 1$ is a coefficient of the memory utilization for data storing,
- f_i is a function of the argument n_i^s that approximates the memory shortage (the size of which is predicted based on [25]) from below for i th resource in comparison with the memory size determined based on testing,
- $d_i^e \geq 0$ is a redundant memory size in GB that can be specified to meet other unexpected memory needs.

We use new parameters f_i and d_i^e in the modified technique. The function f_i is searched for each i th resource individually. The key parameters determined by the specific subject domain are n_i^c , n_i^s , d_i^s , d_i^e , k_i^{ind} , k_i^{bin} , and k_i^{use} . The developer of the scientific application determines these parameters. The parameters d_i^n and k_i^{os} are specified by resource administrators.

Let us demonstrate an example of searching in the function f_i . As i th resource, we consider the first segment of the HPC cluster at the Irkutsk Supercomputer Center [39]. The cluster node has the following characteristics: 2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 64 GB RAM. To evaluate the minimum number of nodes required for the Apache Ignite cluster, we changed the number of scenarios from 27300 to 391300 in increments of 9100. Figure 2 shows the predicted number of nodes.

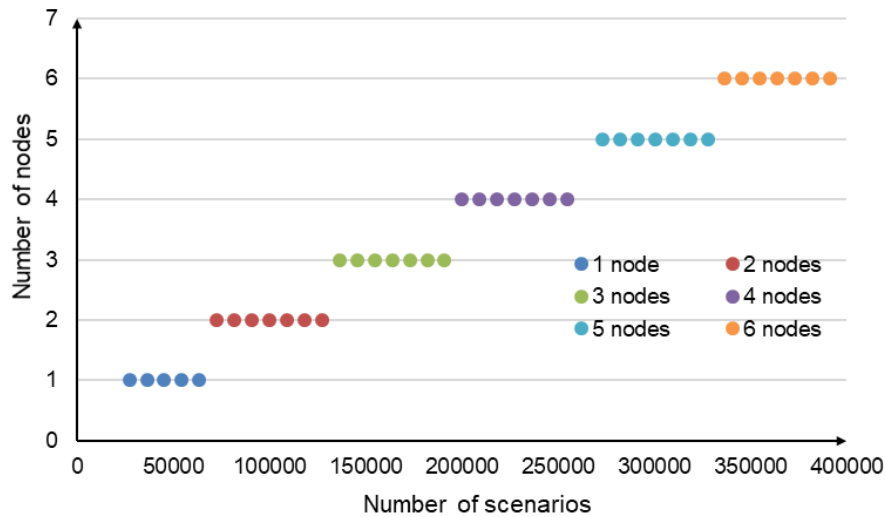


Figure 2. Predicted number of nodes.

Figure 3 shows the deviation of the predicted d_i^r from the actual test required memory size in GB for 1, 2, 3, 4, 5, and 6 nodes of this cluster.

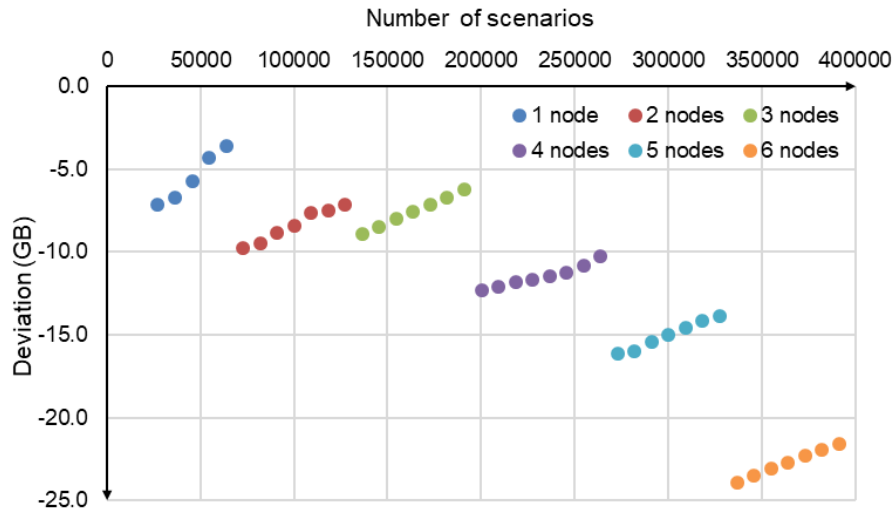


Figure 3. Deviation of predicted data from test data in for different numbers of nodes.

When the Apache Ignite cluster node is launched, Apache Ignite reserves a large number of memory blocks for hash tables. With a small number of disturbance scenarios, the fill density of these blocks is low. Therefore, the memory size per scenario is quite large. As the number of scenarios increases, the fill density of memory blocks grows, and the memory size per scenario decreases (Figure 4). After a new node is added to the Apache Ignite cluster, this process is repeated. This explains the non-monotonic, abrupt, and intermittent nature of the deviation changes in Figure 3.

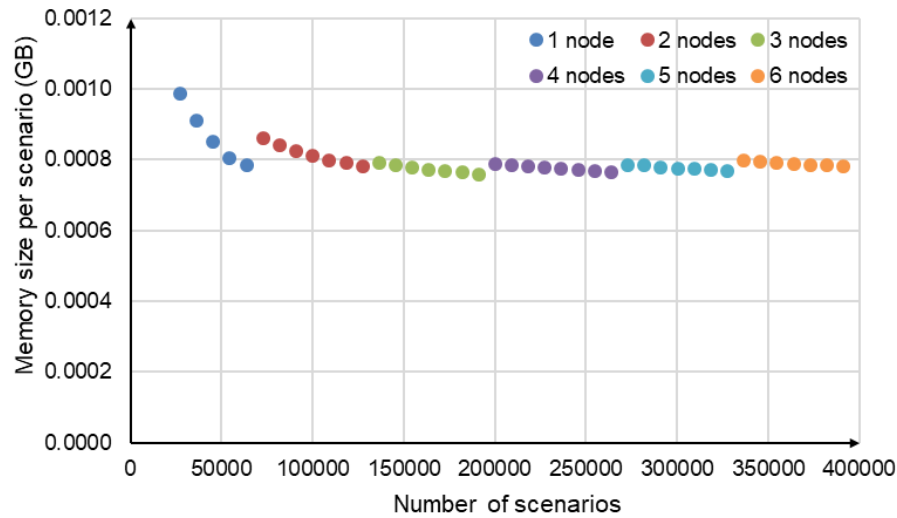


Figure 4. Changing the memory size per scenario.

We search for the linear approximation function $f_i(n_i^s) = a_i n_i^s + b_i$ by solving the following optimization problem using the simplex-method:

$$\sum_{\substack{n_i^s = \overline{n_{i,0}^s}, \overline{n_{i,k}^s} \\ \text{with the step } j}} (f_i^*(n_i^s) - f_i(n_i^s)) \rightarrow \min, \quad (4)$$

$$0 \leq f_i^*(n_i^s) - f_i(n_i^s) \leq 10, \quad (5)$$

where f_i^* is a function of deviations shown in Figure 4, $n_{i,0}^s = 27300$, $n_{i,k}^s = 391300$, $j = 9100$.

Based on the (4) and (5), we obtain the linear function

$$f_i(n_i^s) = -5E - 05n_i^s - 5.852 \quad (6)$$

that approximates the deviations from below (see Figure 5). In Figure 5, $d_1^e = 0$.

Figure 6 demonstrates the approximation error in absolute values that changed from 0 to 10 GB. Therefore, additional research is needed to reduce the observed approximation error in the future. However, applying function (6) provides the necessary evaluation of the required memory to correctly allocate the number of the Apache Ignite cluster nodes.

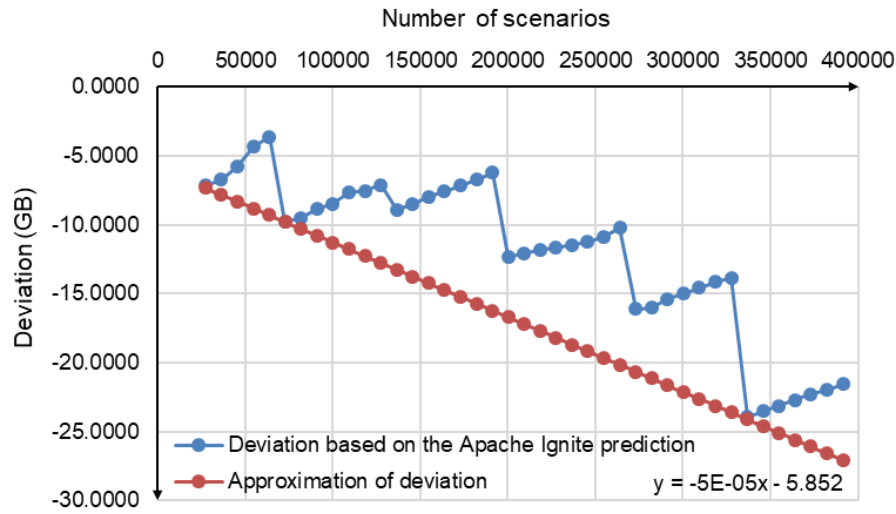


Figure 5. Approximation of deviation.

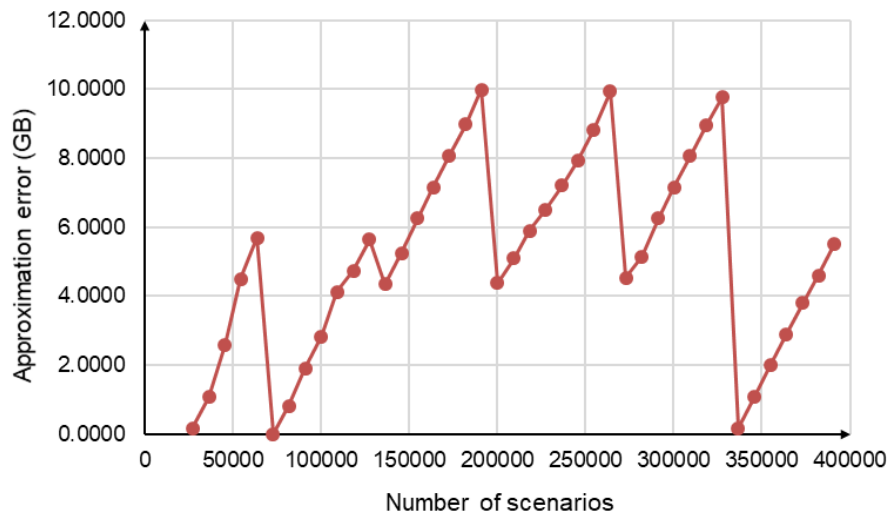


Figure 6. Approximation error.

3.2. Model of the Most Serious Vulnerability

GVA is based on identifying the strongest disturbance scenarios with the most serious consequences for the energy infrastructure. Let $w = (w_1, w_2, \dots, w_{n_w})$ is a vector of vulnerabilities of n_w energy infrastructure configurations. Then, the problem of determining the most severe vulnerability for an energy infrastructure configuration is formulated as follows:

$$w(S, [a, b]) \rightarrow \max, \quad (7)$$

$$g_i(S, Z) \leq 0, \quad i = 1, \dots, n_i, \quad (8)$$

$$h_j(S, Z) = 0, \quad j = 1, \dots, n_j, \quad (9)$$

$$S = \langle G, X, C \rangle, \quad (10)$$

where

- S is a set of the energy infrastructure states on which the disturbance scenarios from Z are imposed,
- a and b are the bounds of a curve segment that reflects the rapid drop of the performance $P(t)$ on time,
- Z is a generated set of disturbance scenarios,
- g_i and h_j are constraints of the energy infrastructure model imposed on S , taking into account the disturbance scenario impact from Z ,
- G and X are the given structure (network) and performance limits of energy infrastructure objects,
- C is environmental conditions.

The solution of the problem (7)-(10) for the configuration is the strongest disturbance scenario found by the brute-force method on Z .

3.3. Algorithms

Let us consider the algorithms used at the preliminary and target computing stages. Algorithms A1 and A2 represent the stage of the preliminary computing. The A1 pseudo code for modeling one series of disturbance scenarios is given below. In the A1 algorithm description, we use the following notations:

- $G = \langle V, U \rangle$ is a directed graph representing the energy infrastructure network, where V is a set of vertices (possible sources, consumers of energy resources, and other objects of energy infrastructure) and U is a set of edges (energy transportation ways),
- n_e is a number of energy infrastructure elements from the set $E = (e_1, e_2, \dots, e_{n_e})$, selected for random failure simulation,
- k is a number of randomly failed elements,
- Z_k is a set of generated disturbances (a set of combinations of n_e by k element failures from E),
- $z_k^i: w \rightarrow \max$ is the strongest disturbance scenario from Z_k for the i th series,
- w_k^i is a set of the most negative changes in the vulnerability corresponding z_k^i for the i th series.

List 1. Algorithm A1.

Inputs: $G, n_e, E = (e_1, e_2, \dots, e_{n_e})$

Outputs: $w_1^i, w_2^i, \dots, w_{n_e}^i$

```

1  begin
2       $k = 1$ ;
3      while  $k \leq n_e$  do
4           $generate(G, E, k \rightarrow Z_k)$ ;
5           $calculate(G, E, k, Z_k \rightarrow z_k^i, w_k^i)$ ;
6           $k = k + 1$ ;
7      end do
9  end

```

The algorithm A2 is briefly described below. It allows us to determine the minimum required number of disturbance scenarios series required to achieve convergence of the GVA results. In the A2 algorithm description, we use the following notations:

- $r_0 \geq 1$ is the initial number of computational experiments,
- q is a number of disturbance scenarios series in each r th computational experiment,
- $\varepsilon \geq 0$ is the given computation accuracy,

- l is the minimum required number of disturbance scenarios series to perform GVA,
- \tilde{p}_k^r is the average performance value in the computational experiment r on the k th disturbance scenario,
- $\Delta\tilde{p}_k^{r,r-1}$ is the difference between $(r-1)$ th and r th computational experiments on the k th disturbance scenario,
- $\overline{\Delta\tilde{p}_k^{r,r-1}}$ is the average (statistical mean) of the set of values $\Delta\tilde{p}_k^{r,r-1}$,
- $\sigma^{r,r-1}$ is the standard deviation of the set of differences $\Delta\tilde{p}_k^{r,r-1}$.

List 2. Algorithm A2.

Inputs: $G, n_e, E = (e_1, e_2, \dots, e_{n_e}), r_0, q, \varepsilon$

Outputs: l

```

1  begin
2       $r = r_0;$ 
3      for  $i = \overline{1, rq}$  step 1 do
4           $A1(G, n_e, E \rightarrow w_1^i, w_2^i, \dots, w_{n_e}^i);$ 
5      end do
6      for  $k = \overline{1, n_e}$  step 1 do
7           $\tilde{p}_k^r = \frac{1}{q} \sum_{i=1}^q w_k^i;$ 
8      end do
9      if  $r = r_0$  then  $r = r + 1;$  goto 3
10     for  $k = \overline{1, n_e}$  step 1 do
11          $\Delta\tilde{p}_k^{r,r-1} = |\tilde{p}_k^r - \tilde{p}_k^{r-1}|;$ 
12     end do
13      $\sigma^{r,r-1} = \sqrt{\frac{\sum_{k=1}^{n_e} (\Delta\tilde{p}_k^{r,r-1} - \overline{\Delta\tilde{p}_k^{r,r-1}})^2}{n_e}};$ 
14     if  $\sigma^{r,r-1} \geq \varepsilon$  then  $r = r + 1;$  goto 3;
15      $l = rq;$ 
16 end

```

The primary purpose of the target computing stage is to plot the dependence of the average performance value \tilde{p}_k on the k failed elements of the considered energy infrastructure. The algorithm A3 represents the target computing stage. In the A3 algorithm description, we use the following notations:

- $\hat{q} \gg l$ is a number of disturbance scenarios series,
- $plot$ is a plot based on $\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_{\hat{q}}$.

List 3. Algorithm A3.**Inputs:** $G, n_e, E = (e_1, e_2, \dots, e_{n_e}), \hat{q}$ **Outputs:** $plot$

```

1  begin
2      for  $i = \overline{1, \hat{q}}$  step 1 do
3           $A1(G, n_e, E \rightarrow w_1^i, w_2^i, \dots, w_{n_e}^i);$ 
4      end do
5      for  $k = \overline{1, n_e}$  step 1 do
6           $\tilde{p}_k = \frac{1}{\hat{q}} \sum_{i=1}^{\hat{q}} w_k^i;$ 
7      end do
8       $plotting(\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_{n_e} \rightarrow plot);$ 
9  end

```

4. Application

We use the OT framework to develop and apply a scientific application for studying the resilience of energy infrastructures. The first version of this application includes workflows for studying the vulnerability as a key metric of the resilience.

OT includes the following main components:

- User interface,
- Designer of a computational model that implements the knowledge specification about the subject domain of solved problems,
- Converter of subject domain descriptions (an auxiliary component of the computational model designer), represented in domain-specific languages, into the computational model,
- Designer of module libraries (auxiliary component of the computational model designer) that support the development and modification of applied and system software,
- Software manager that provides continuous integration of applied and system software,
- Computation manager that implements computation planning and resource allocation in a heterogeneous distributed computing environment,
- Computing environment state manager that observes resources,
- API for access to external information, computing systems, and resources, including digital platform of ecological monitoring of BNT [40],
- Knowledge base about computational model and computation databases with the testing data, initial data, and computing results.

The conceptual model contains information about the scientific application, which has a modular structure. The model specifies sets of parameters, abstract operations on the parameter field, applied and system software modules that implement abstract operations, workflows, and relations between the listed objects. In addition, this model describes the hardware and software infrastructure (characteristics of nodes, communication channels, network devices, network topology, etc.). OT provides the development of system modules that implement computing environment monitoring, interacting with meta-schedulers, local resource managers, and IMDG middleware, transferring, pre- and post-processing of data, etc. System operations and modules can be included in the computational model and executed within workflows.

End-users of the application execute workflows to perform computational experiments. The computation manager controls the workflow executions in the computing environment. It receives information about the state of computing processes and resources from the manager of the computing environment state. In addition, the computation manager can visualize the calculated parameters and publish workflows as WPSs on the digital platform of ecological monitoring using the API mentioned above.

Figure 7 shows the structure of the HPC environment. The debugged and tested applied software within the testing stage is deployed on resources (computational clusters) of the environment. At the preliminary and target stages, the subject domain expert interacts with the OT components. He sets initial data (energy infrastructure configurations and key application parameters), executes application workflows, controls their completion, and obtains and evaluates computation results. Workflows for GVA are shown in Figures 8–10. The Inputs and Outputs elements represent the input and output data of workflows, respectively.

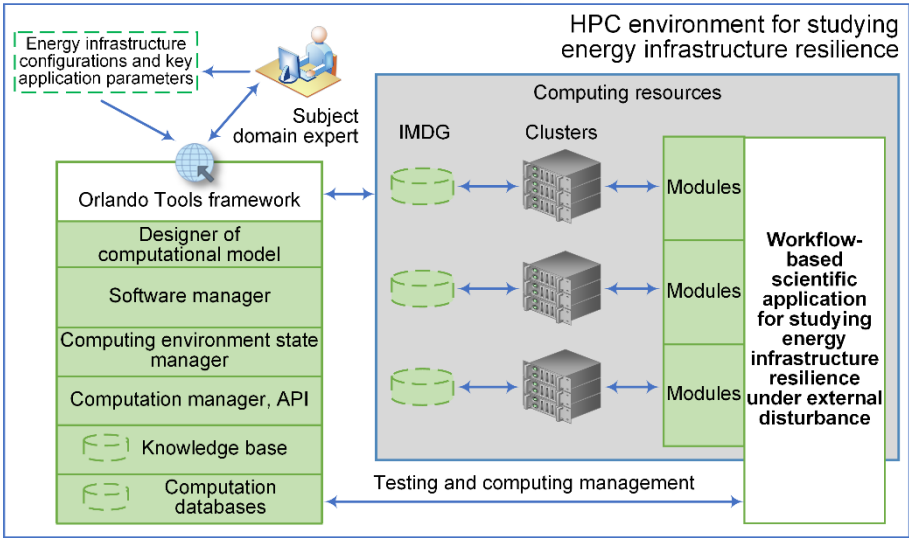


Figure 7. HPC environment structure.

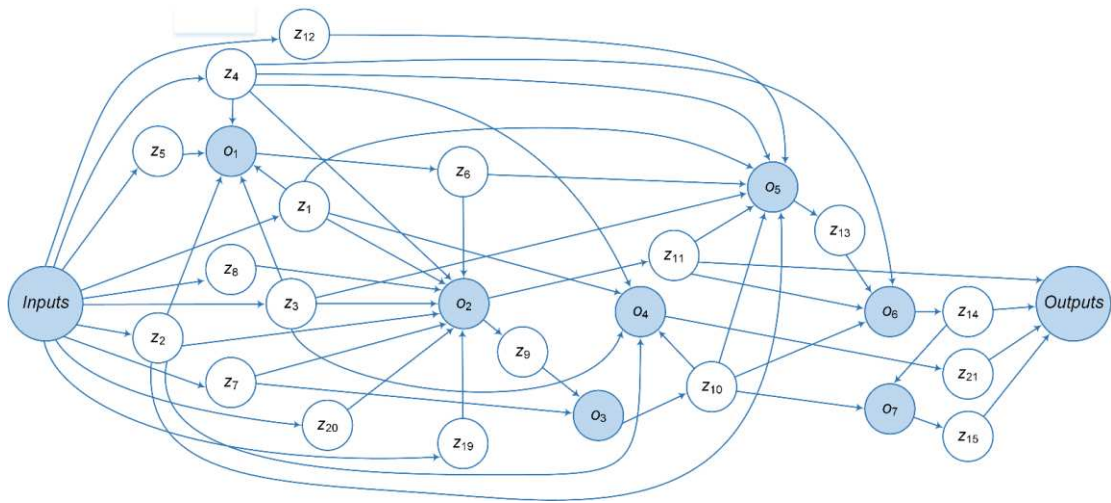


Figure 8. Workflow 1.

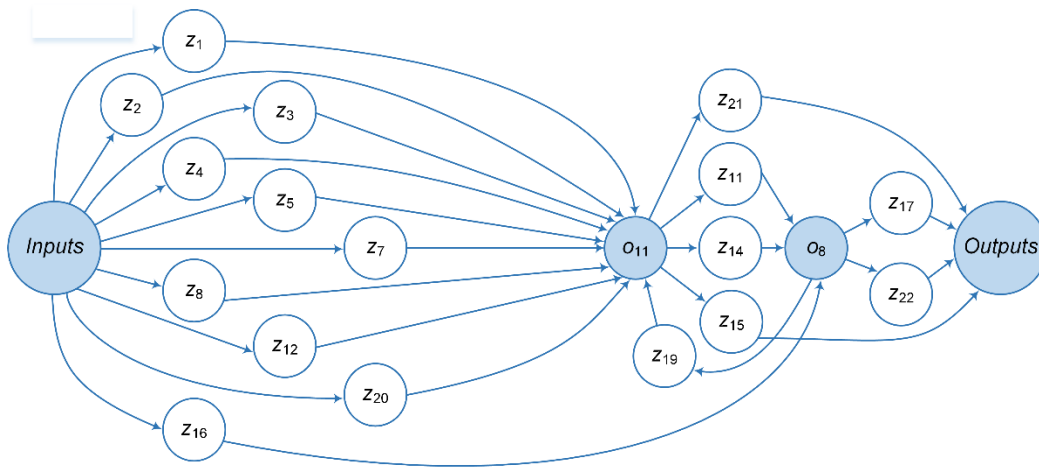


Figure 9. Workflow 2.

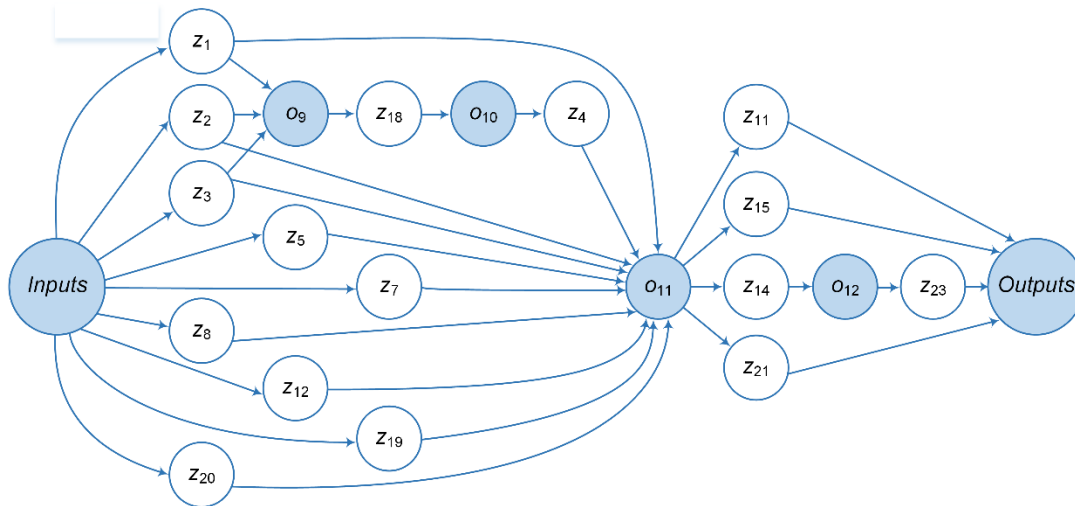


Figure 10. Workflow 3.

Workflow 1 computes the decrease in the energy system performance with the given configuration due to mass failures of its elements. Operation o_1 retrieves a list of energy system elements from the computation database. Then, operation o_2 determines the required number of nodes to form an Apache Ignite IMDG cluster and configures its nodes. Next, operation o_3 launches an instance of the IMDG cluster, to which operation o_4 writes the structure of the energy system. Operation o_5 models a series of disturbances. The results of its execution are aggregated by the operation o_6 . IMDG cluster nodes are stopped and released by the operation o_7 .

Workflow 2 implements an algorithm for searching the minimum number of disturbance scenarios series required to obtain a stable evaluation of the decrease in the energy system performance. It is executed within the preliminary computing stage and applied to one energy infrastructure configuration selected by the subject domain specialist. Operation o_{11} reflects Workflow 1. Operation o_8 checks whether the required minimum number of disturbance scenarios has been achieved. If the condition $(z_{11} \neq z_{20}) \wedge (z_{17} < z_{16})$ is true, then Workflow 2 is completed. Otherwise, r is incremented by one and a new experiment is performed.

Workflow 3 realizes the target computing stage. Operation o_9 extracts a set of configurations of the energy system under study from the computation database. Operation o_{10} decomposes the general problem into separate subproblems, where one configuration is processed in parallel by an instance of operation o_{11} . The input parameter z_{20} has the initial value equal \hat{q} . Finally, operation o_{12} draws the plot.

The parameters $z_1 - z_{22}$ are inputs and outputs of operations, where:

- z_1, z_2 , and z_3 determine the graph G ,
- z_4 is an energy infrastructure configuration ID,
- z_5 is a type of energy infrastructure elements,
- z_6 represents the set E and the number n_e of its elements,
- z_7 is a set of options for the Apache Ignite cluster,
- z_8 is a list of d_i^n , $i = \overline{1, n_r}$,
- z_9 is a list of n_i , $i = \overline{1, n_r}$,
- z_{10} is a list of node network addresses of the Apache Ignite cluster,
- z_{11} and z_{12} represent r and q correspondingly,
- z_{13} is a list of w_k^i , $i = \overline{1, r}, k = \overline{1, n_e}$,
- z_{14} is a list of \tilde{p}_k^r for r th computational experiment, $k = \overline{1, n_e}$,
- z_{15} is the exit code of the operation o_7 ,
- z_{16} and z_{17} specify ε and s correspondingly,
- z_{18} is a list of energy infrastructure configuration IDs,
- z_{19} and z_{20} determine $r + 1$ and r_0 correspondingly,
- z_{21} is a log file,
- z_{22} represents l ,
- z_{23} is a plot.

To distribute a set of scenarios among the resources that are available to run the experiment, the OT computation manager solves the following optimization problem:

$$\tau = \max_{i=\overline{1, n_r}}(\mu_i) \rightarrow \min, \quad (11)$$

$$\mu_i = \epsilon_i + \frac{v_i}{\eta_i \pi_i}, \quad (12)$$

$$\sum_{i=1}^{n_r} v_i = n_s, \quad 0 \leq \mu_i \leq \delta_i, \quad \eta_i^{\min} \leq \eta_i \leq \lambda_i, \quad (13)$$

where

- ϵ_i is overheads for the use of i th resource independent of the number of scenarios,
- v_i is a number of scenarios under processing on i th resource,
- η_i is the number of nodes dedicated to i th resource,
- π_i is a performance of i th resource defined by the ratio of the number of scenarios per unit of time,
- δ_i is a quota on time-of-use of i th resource for problem-solving,
- λ_i is a quota on a number of nodes of i th resource for problem-solving,
- $\eta_i^{\min} \geq 1$ is the minimum number of nodes of i th resource taking into account the required memory size for IMDG.

In this problem formulation (11)-(13), minimization ensures a rational use of resources and balancing their computational load. The resources used may have different computational characteristics of their nodes. Redistribution for part of the computing load from faster to slower nodes can be encouraged if the makespan of the workflow execution is reduced.

5. Computational Experiments

We performed GVA for an energy infrastructure model similar to one of the segments for the gas transportation network in Europe and Russia. This model includes 910 elements, including 332 nodes (28 sources, 64 consumers, 24 underground storage facilities, and 216 compressor stations) and 578 sections of main pipelines and branches of distribution networks. The total amount of natural gas supplied to consumers was chosen as the performance measure $P(t)$ for this segment of the gas

supply network. The measure $P(t)$ is normalized by converting the resulting supply value into a percentage of the required total energy demand following [4].

Computational experiments were performed in the heterogeneous distributed computing environment on the following resources:

- HPC Cluster 1 with the following node characteristics: 2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 64 GB RAM,
- HPC Cluster 2 with the following node characteristics: 2x18 cores CPU Intel Xeon X5670, 2.1 GHz, 128 GB RAM,
- HPC Cluster 3 with the following node characteristics: 2x64 cores CPU Kunpeng-920, 2.6 GHz, 1 TB RAM.

Below are the results of the preliminary and target computing.

The search for the minimum number of series of disturbance scenarios is performed by increasing the initial value of the number of sequences. We increase the number of sequences until the convergence of the results of two GVA iterations, determined by the value of ε , is achieved. Due to the high degree of uncertainty in the data characterizing the behavior of the selected gas supply network under conditions of major disturbances, we use ε equal to 0.025%.

We started the preliminary computing with $r_0 = 1$ and $q = 5$. The minimum required number of disturbance scenario series was found at $r = 200$. Therefore, $l = 1000$. Next, according to the GVA methodology [6], we set \hat{q} equals to 2000 and r equals to 400 to perform the target computing.

The convergence process for the experimental results is illustrated in Figures 11–13. These figures show the graphs of $\Delta\tilde{p}_k^{12,1}$, $\Delta\tilde{p}_k^{200,12}$, and $\Delta\tilde{p}_k^{400,200}$, which reflect the difference in performance degradation between the following pairs of computational experiments: $(r = 1, r = 12)$, $(r = 12, r = 200)$, and $(r = 200, r = 400)$. The difference in the consequences of large disturbances is given on the y -axis. The x -axis represents the amplitude of large disturbances, expressed as a percentage of the number of failed elements. In addition, the standard deviations $\sigma^{12,1} = 0.028$, $\sigma^{200,12} = 0.021$, and $\sigma^{400,200} = 0.009$ are shown in the upper right corners of the corresponding figures.

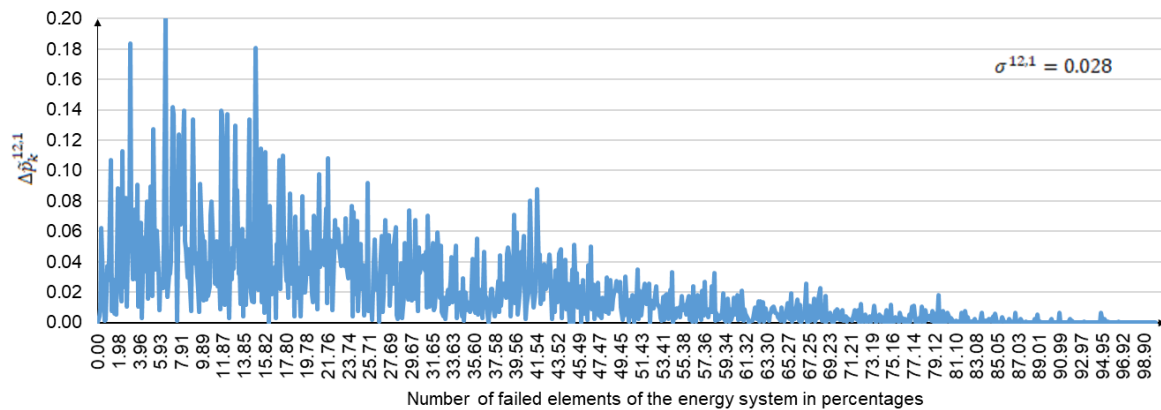


Figure 11. Graph of $\Delta\tilde{p}_k^{12,1}$.

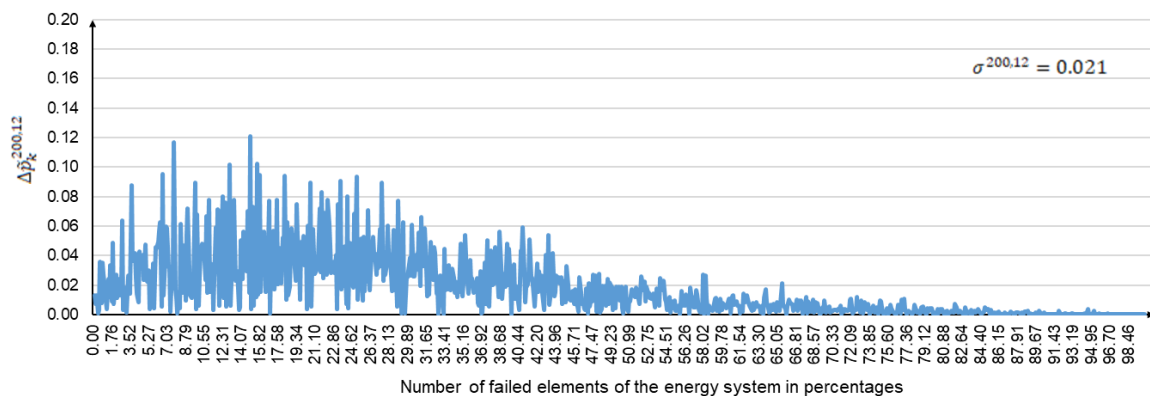
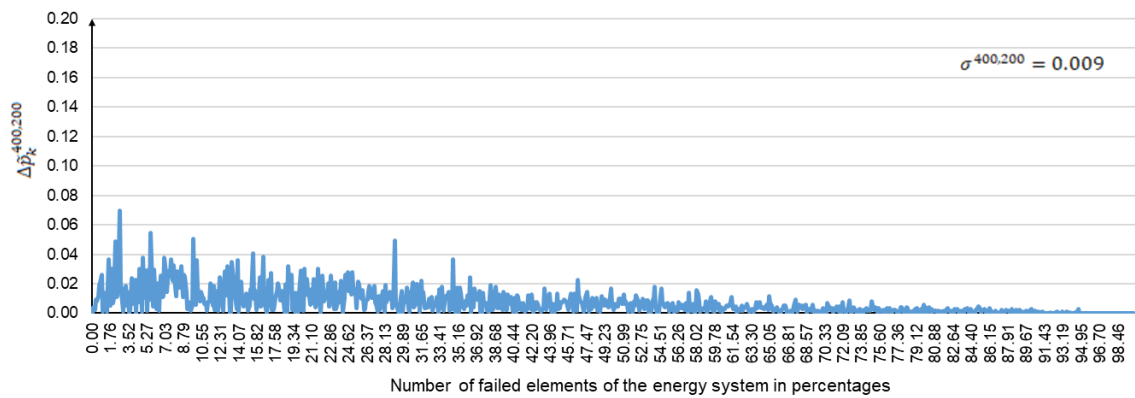


Figure 12. Graph of $\Delta \tilde{p}_k^{200,12}$.**Figure 13.** Graph of $\Delta \tilde{p}_k^{400,200}$.

Figures 11–13 show that the largest scatters in the difference between the consequences of large disturbances occur at the initial values of k . They then decrease as k increases. The magnitude of the dispersion is determined by the criticality of the network elements, whose failures are modeled in disturbance scenarios. This is especially noticeable for small values of k . The consequences of failures of some critical elements for a selected segment of the natural gas transportation network can reach 10% or more. At the same time, failures of most non-critical elements do not significantly affect the supply of natural gas to consumers. As k increases, the dependence on the criticality of failed network elements decreases. This leads to a decrease in the spread of the difference in the consequences of large disturbances. In general, the above results of target computing are consistent with the results in [6].

Figure 14 shows the drop in the performance for the energy system segment under consideration. The drop is shown depending on the number of its failed elements for various failures. We can see that serious problems in gas supply to consumers begin already with a small number of failed elements, which is about 3%. Here, a serious problem in gas supply means a drop in the performance by 20% or more.

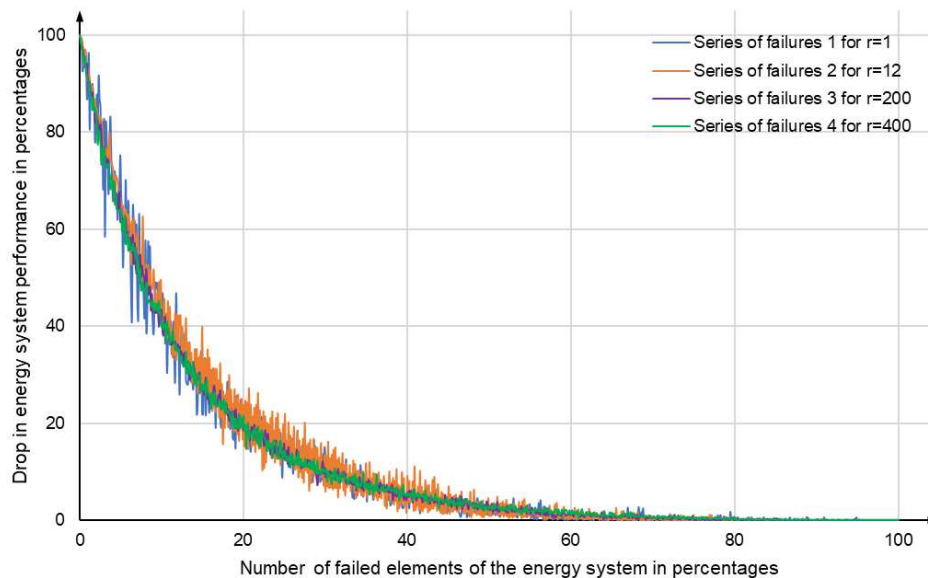
**Figure 14.** Decreasing the energy system performance.

Table 3 provides several important quantitative characteristics of GVA for various technical systems [41–47], including the considered segment of the gas supply system. A more detailed

comparison in dependences of system vulnerabilities on disturbance amplitudes is difficult to perform even at a qualitative level. This is due to the following reasons.

- There are differences in the types of systems under study and the details of representing the distribution of resource flows across systems in their models. For example, for simple linear models of railway transport [43], gas supply [44,45] and electricity supply [41,46,47], it is sufficient to use only information about the capacity of system network edges. At the same time, the nonlinear water supply model requires additional considerations of the pressure in nodes (network vertices) and the hydraulic resistance of edges. In [47], in contrast to [41,46], the additional possibility of cascading propagation of accidents throughout the system during the impact of a major disturbance is taken into account. In our study, we use a gas supply model similar to [44,45].
- The network structure of a specific system configuration largely determines its vulnerability. Xie et al. [47] clearly showed that the vulnerability dependencies on the amplitude of disturbances are significantly different for two different electrical power systems.
- Various principles of random selection of elements for GVA are used. In our work as well as in [41], all network elements (vertices and edges) can be selected for failure (see Table 3). At the same time, in [42,43,45,46], only edges are selected. In [44,46], only edges with low centrality values are selected for failure.

Overall, our approach to GVA [48] integrates and develops the advantages of the approaches presented in [6,41]. In particular, we combine models of different energy systems to study their joint operation under extreme conditions. Such combination is based on the unified principles of interaction with models of energy infrastructures at different levels of the territorial-sectoral hierarchy.

Our methodological result is that we have ensured that all elements (vertices and edges) of the gas transportation network can be included in the set of failed elements. This allowed us to perform a complete analysis of the production and transportation subsystems of the gas supply system in contrast to [44,45], where only edge failures were modeled. The possible synergistic effect of simultaneous failures of several network elements is important. It lies in the fact that the consequences of a group failure can be higher than the sum of the consequences of individual failures. The synergetic effect affects the curves presented in Figure 14, increasing their slope. As the search for critical elements showed in [26], the synergetic effect appears only when considering joint failures at the vertices and edges of the energy network. The steepness of the curves in Figure 14, which is greater than that of similar curves in [39], is due precisely to the synergetic effect. The system vulnerability analysis performed in [43–46] also does not consider the synergistic effect of simultaneous failures at the vertices and edges of the energy network.

Table 3. Characteristics of experiments.

System	Literary Source	Resource	Number of Vertices	Number of Edges	Maximum Number of Simultaneously Failed Elements	Failed Elements
A small town power system	[41]	Power	352	452	804	All vertices and edges
Net3	[42]	Water	97	119	119	All edges
OBCL-1	[42]	Water	263	289	289	All edges
D-town	[42]	Water	407	459	459	All edges
Ky14	[42]	Water	384	553	553	All edges
Ky3	[42]	Water	275	371	371	All edges
80 virtual networks	[42]	Water	102–506	110–554	554	All edges
Our infrastructure	–	Natural gas	332	578	910	All vertices and edges

Soviet Rail system circa 1955	[43]	Railway	50	89	7	Edges
Provincial pipeline network	[44]	Natural gas	35	34	10	Edges
A transmission network	[45]	Natural gas	54	70	70	All edges
IEEE 118 bus system	[46]	Power	118	186	10	Edges
IEEE 118 bus system	[47]	Power	118	186	30	Vertices and edges
IEEE 30 bus system	[47]	Power	30	41	10	Vertices and edges

Moreover, the size of the studied segment of the gas supply system significantly exceeds similar characteristics of the experiments represented in [42–47]. The size of the network determines the maximum number of simultaneous random failures, which is necessary to construct the dependencies in Figure 14. Xie et al. showed that 10 random attacks in [46] are not enough to conclude about the stability of an electric power system of similar size to the IEEE 118 bus system. In our experiment, the maximum number of simultaneously failed elements is equal to the number of network elements. This allows us to draw conclusions about the boundary beyond which the system collapses, i.e., breaks down into independent parts. A model with a high quality and quantity of the input data delivers more founded outputs.

In Table 4, we show the experimental results for HPC Cluster 1 ($i = 1$), HPC Cluster 2 ($i = 2$), and HPC Cluster 3 ($i = 3$). In GVA problem-solving, we have processed 1820000 disturbance scenarios. The values of η_i , v_i , and μ_i are calculated based on solving the problem (11)–(13) taking into account δ_i , λ_i , and ϵ_i . μ_i reflects the predicted Data Processing Time (DPT) on η_i nodes of i th HPC cluster. The parameter η_i^{min} is determined using (1)–(3). It sets the minimum number of nodes required for i th Apache Ignite cluster to provide computation reliability, $\eta_i^{min} \leq \eta_i$. We add $\eta_i - \eta_i^{min}$ nodes on i th HPC cluster, $i = \overline{1, n_r}$. This provides balancing the computing load in proportion to the node performance and decreasing the real DPT on η_i nodes in comparison with the predicted DPT on η_i^{min} nodes. The addition of the nodes is made within the existing quotas λ_i . The error of μ_i (the DPT prediction error) does not exceed 8.15% compared to the DPT obtained in computational experiments on η_i nodes.

Table 4. Data processing time and computation speedup.

i	δ_i (s)	ϵ_i (s)	λ_i	η_i	v_i	μ_i	η_i^{min}	DPT on η_i^{min} nodes (s)	DPT on η_i nodes (s)	Error	Computation Speedup
1	1728000.00	120	15	15	216230	736.61	6	1561.53	711.61	3.39	2.24
2	864000.00	100	20	20	1312587	736.63	15	895.51	694.83	5.67	1.29
3	1296000.00	90	3	3	291183	736.63	1	1909.89	676.63	8.15	2.82

Figure 15 shows the computation speedup achieved by adding new Apache Ignite nodes on the HPC clusters. The speedup increases with the following increases in the number of nodes:

- From 6 to 15 nodes for HPC Cluster 1,
- From 15 to 20 nodes for HPC Cluster 2,
- From 1 to 3 nodes for HPC Cluster 3.

We see that the makespan T is reduced by more than two times (from 1909.89 to 711.61) compared to the makespan obtained with the number of nodes required by Apache Ignite for a given problem where

$$T = \begin{cases} \max_{i=1, n_r} \eta_i^{min} & \text{for DPT on } \eta_i^{min} \text{ nodes,} \\ \max_{i=1, n_r} \eta_i & \text{for DPT on } \eta_i \text{ nodes.} \end{cases}$$

In all three cases, the speedup is close to linear.

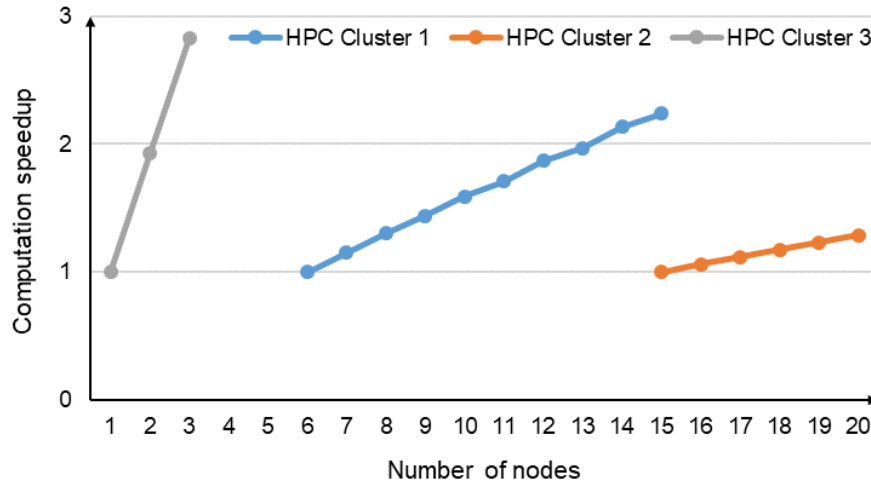


Figure 15. Computation speedup for the HPS clusters.

6. Discussion

Nowadays, studying the development and use of energy systems in terms of environmental monitoring and conservation is undoubtedly a challenge. Resilience of energy systems is one of the applications within such a study. In terms of maintaining a friendly environmental situation, conserving natural resources, and ensuring balanced energy consumption, increasing the resilience of energy systems can prevent the negative consequences of significant external disturbances.

Top-down, bottom-up, and hybrid approaches to energy modeling can be implemented at macro and micro levels with low, medium, and high data requirements. In addition, energy system models can include one sector, such as the electricity sector, or multiple sectors and assume optimization, agent-based, stochastic, indicator-based, or hybrid modeling. In all cases, large-scale experiments are considered to be repeated.

Preparing large-scale experiments for this study is quite a long and rigorous work. Within such a work, it is necessary to consider as much as possible the subject domain specificity and the end-user requirements concerning the computing environment used.

This requires the development and application of specialized tools to support different aspects of studying the resilience of energy systems. These aspects include the presence of large data sets, need to speed up their processing, demand for HPC use, convergence of applied and system software, provision of flexible and convenient service-oriented of end-user access to the developed models and algorithms, etc. Therefore, it is evident that system models, algorithms, and software are required to the same extent as applied developments to provide efficient problem-solving and rational resource utilization within large-scale experiments.

Unfortunately, there are no ready-made solutions in the field of resilience research for its different metrics. To this end, we focus on designing a new approach to integrate workflow-based applications with IMDG technology and WPSs to implement the resilience study for energy systems in the HPC environment.

7. Conclusions

In this paper, we propose a new approach to implement HPC-based analysis of the resilience of workflow-based energy infrastructures, focusing on their vulnerability.

Compared to similar solutions, this approach studies large energy infrastructures, modeling the number of simultaneously failed elements of different types (vertices and edges in the infrastructure network) up to the number of network elements. It reduces the problem-solving makespan by more than twice. Furthermore, it achieves near-linear speedup in each resource.

In addition, we develop and provide:

- Workflow-based scientific applications to study the resilience of energy infrastructures, providing WPS-oriented access to problem-solving in the geosciences;
 - Automatic transformation of subject domain descriptions, expressed in domain-specific languages, into a specialized computational model;
 - Ability to create a computing environment based on the integration of heterogeneous resources, such as HPC clusters at public access supercomputer centers, Grid systems, cloud platforms, and specialized end-user computers and servers;
 - System parameters and operations support the workflows to interact with both the computing environment and the IMDG middleware;
 - Automate dynamic deployment and configuration of the IMDG clusters to available nodes within the workflow execution;
 - Reliable computation based on testing the applied software and allocation of available nodes
- Future work will focus on the following research directions:
- Expanding the library of scientific applications by creating new workflows to study other resiliency metrics;
 - Optimizing the prediction of the required memory for data processing on an Apache Ignite cluster;
 - Conducting large-scale experiments to study resilience concerning a complete set of metrics on existing infrastructures of fuel and energy complexes and their components.

Author Contributions: Conceptualization and methodology, A.F. and A.E.; software, A.E. and S.G.; validation, A.F., A.E., S.G. and O.B.; formal analysis, A.F. and A.E.; investigation, A.E.; resources, A.F. and E.F.; data curation, A.E., O.B. and E.F.; writing—original draft preparation, A.F., A.E., A.T., S.G., O.B. and E.F.; writing—review and editing, A.F., A.E., A.T., S.G., O.B. and E.F.; visualization, A.E. and O.B.; supervision, A.F. and A.T.; project administration, A.F. and A.T.; funding acquisition, A.F. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Ministry of Science and Higher Education of the Russian Federation, grant no. 075-15-2020-787 for implementation of Major scientific projects on priority areas of scientific and technological development (the project «Fundamentals, methods and technologies for digital monitoring and forecasting of the environmental situation on the Baikal natural territory»).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zio, E. Challenges in the vulnerability and risk analysis of critical infrastructures. *Reliab. Eng. Syst. Safe.* **2016**, *152*, 137–150, doi:10.1016/j.res.2016.02.009
2. Ahmadi, S.; Saboohi, Y.; Vakili, A. Frameworks, quantitative indicators, characters, and modeling approaches to analysis of energy system resilience: A review. *Renew. Sustain. Energy Rev.* **2021**, *144*, 110988, doi:10.1016/j.rser.2021.110988
3. Voropai, N.; Rehtanz, C. Flexibility and Resiliency of Electric Power Systems: Analysis of Definitions and Content. In Proceedings of the EPJ Web of Conferences, Irkutsk, Russia, 26–31 August 2019, doi:10.1051/epjconf/201921701018
4. Poulin, C.; Kane, M.B. Infrastructure resilience curves: Performance measures and summary metrics. *Reliab. Eng. Syst. Safe.*, **2021**, *216*, 107926, doi:10.1016/j.res.2021.107926
5. Wang, W.; Zhang, Y.; Li, Y.; Hu, Q.; Liu, C.; Liu, C. Vulnerability analysis method based on risk assessment for gas transmission capabilities of natural gas pipeline networks. *Reliab. Eng. Syst. Safe.* **2022**, *218*, 108150, doi:10.1016/j.res.2021.108150

6. Mugume, S.N.; Gomez, D.E.; Fu, G.; Armani, R.; Butler, D. A global analysis approach for investigating structural resilience in urban drainage systems. *Water res.* **2015**, *81*, 15–26, doi:10.1016/j.watres.2015.05.030
7. Dobson, I.; Carreras, B.A.; Lynch, V.E.; Newman, D.E. Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization. *Chaos An Interdis. J. Nonlinear Sci.* **2007**, *17*(2), 026103, doi:10.1063/1.2737822
8. Gorton, I.; Huang, Z.; Chen, Y.; Kalahar, B.; Jin, S. A high-performance hybrid computing approach to massive contingency analysis in the power grid. In Proceedings of the 5th IEEE International Conference on e-Science, Oxford, Great Britain, 9-11 December 2009, doi:10.1109/e-Science.2009.46
9. Zhang, J.; Razik, L.; Jakobsen, S.H.; D'Arco, S.; Benigni, A. An Open-Source Many-Scenario Approach for Power System Dynamic Simulation on HPC Clusters, *Electronics*. **2021**, *10*(11), 1330, doi:10.3390/electronics10111330
10. Khaitan, S.K. A survey of high-performance computing approaches in power systems. In Proceedings of the IEEE Power and Energy Society General Meeting, USA, Boston, 17-21 July 2016, doi:10.1109/PESGM.2016.7741984
11. Anderson, E.J.; Linderoth, J. High throughput computing for massive scenario analysis and optimization to minimize cascading blackout risk. *IEEE T. Smart Grid.* **2016**, *8*(3), 1427–1435, doi:10.1109/TSG.2016.2646640
12. Liew, C.S.; Atkinson, M.P.; Galea, M.; Ang, T.F.; Martin, P.; Hemert, J.I.V. Scientific workflows: moving across paradigms. *ACM Comput. Surv.* **2016**, *49*(4), 1–39, doi:10.1145/3012429
13. Guroob, A.H. EA2-IMDG: Efficient Approach of Using an In-Memory Data Grid to Improve the Performance of Replication and Scheduling in Grid Environment Systems. *Computation*. **2023**, *11*(3), 65, doi:10.3390/computation11030065
14. Grover, P.; Kar, A.K. Big data analytics: A review on theoretical contributions and tools used in literature. *Global Journal of Flexible Systems Management*. **2017**, *18*, 203–229, doi:10.1007/s40171-017-0159-3
15. de Souza Cimino, L.; de Resende, J.E.E.; Silva, L.H.M.; Rocha, S.Q.S.; de Oliveira Correia, M.; Monteiro, G.S.; de Souza Fernandes, G.N.; da Silva Moreira, R.; de Silva, J.G.; Santos, M.I.B.; Aquino, A.L.L.; Almeida, A.L.B.; de Castro Lima, J. A middleware solution for integrating and exploring IoT and HPC capabilities. *Softw. Pract. Exper.* **2019**, *49*(4), 584–616, doi:10.1002/spe.2630
16. Hazelcast. Available online: <https://hazelcast.com/> (accessed on 22 September 2023).
17. Infinispan. In-Memory Distributed Data Store. Available online: <https://infinispan.org/> (accessed on 22 September 2023).
18. Apache Ignite. Distributed Database for High-Performance Applications with In-Memory Speed. Available online: <https://ignite.apache.org/> (accessed on 22 September 2023).
19. Johns, M. Getting Started with Hazelcast. Packt Publishing Ltd: Birmingham, UK, 2013; ISBN 9781783554058, 1783554053. Available online: <https://www.programmer-books.com/wp-content/uploads/2020/01/Getting-Started-with-Hazelcast.pdf> (accessed on 22 September 2023).
20. Marchioni, F. Infinispan Data Grid Platform. Packt Publishing Ltd: Birmingham, UK, 2012 ISBN 184951822X, 9781849518222.
21. Bhuiyan S.A.; Zheludkov M.; Isachenko T. High Performance in-memory computing with Apache Ignite. Leanpub: Victoria, British Columbia, Canada, 2018. Available online: <http://samples.leanpub.com/ignite-sample.pdf> (accessed on 22 September 2023).
22. Kathiravelu, P.; Veiga, L. An adaptive distributed simulator for cloud and mapreduce algorithms and architectures. In Proceedings of the 7th International Conference on Utility and Cloud Computing, London, United Kingdom, 8-11 December 2014, doi:10.1109/UCC.2014.16
23. Zhou, M.; Feng, D. Application of in-memory computing to online power grid analysis. *IFAC-PapersOnLine*. **2018**, *51*(28), 132–137, doi:10.1016/j.ifacol.2018.11.690
24. Zhou, M.; Yan, J.; Wu, Q. Graph Computing and Its Application in Power Grid Analysis. *CSEE J. Power Energy Syst.* **2022**, *8*(6), 1550–1557, doi:10.17775/CSEEJPES.2021.00430
25. Capacity Planning. Available online: <https://www.gridgain.com/docs/latest/administrators-guide/capacity-planning> (accessed on 22 September 2023).
26. Gorsky, S.; Edelev, A.; Feoktistov, A. Data Processing in Problem-Solving of Energy System Vulnerability Based on In-Memory Data Grid. *Lect. Notes Net. Sys.* **2022**, *424*, 271–279, doi: 10.1007/978-3-030-97020-8_25
27. Feoktistov, A.; Gorsky, S.; Sidorov, I.; Bychkov, I.; Tchernykh, A.; Edelev, A. Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment. *Commun. Comput. Inf. Sci.* **2020**, *1087*, 18–32, doi: 10.1007/978-3-030-41005-6_2
28. Magazzino, C.; Mele, M.; Schneider, N. A machine learning approach on the relationship among solar and wind energy production, coal consumption, GDP, and CO2 emissions. *Renew. Energ.* **2021**, *167*, 99–115, doi:10.1016/j.renene.2020.11.050
29. UNICORE. Available online: <https://www.unicore.eu/> (accessed on 22 September 2023).
30. DAGMan. Available online: <https://htcondor.org/dagman/dagman.html> (accessed on 22 September 2023).

31. Deelman, E.; Singh, G.; Su, M.H.; Blythe, J.; Gil, Y.; Kesselman, C.; Mehta, G.; Vahi, K.; Berriman, G.B.; Good, J.; Laity, A.; Jacob, J.C. Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems. *Sci. Programming-Neth.* **2005**, *13*(3), 219–237, doi:10.1155/2005/128026
32. Balis, B. HyperFlow: A model of computation, programming approach and enactment engine for complex distributed workflows. *Future Gener. Comp. Sy.* **2016**, *55*, 147–162, doi:10.1016/j.future.2015.08.015
33. Hilman, M.H.; Rodriguez, M.A.; Buyya, R. Workflow-as-a-service cloud platform and deployment of bioinformatics workflow applications. In Knowledge Management in the Development of Data-Intensive Systems; Mistrik, I., Galster, M., Maxim, B., Tekinerdogan B., Eds.; CRC Press: Boca Raton, Florida, US, 2021; pp. 205–226.
34. Goecks, J.; Nekrutenko, A.; Taylor, J.; Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome boil.* **2010**, *11*, 1–13, doi:10.1186/gb-2010-11-8-r86
35. Yue, P.; Zhang, M.; Tan, Z. A geoprocessing workflow system for environmental monitoring and integrated modelling. *Environ. Modell. Softw.* **2015**, *69*, 128–140, doi: 10.1016/j.envsoft.2015.03.017
36. Tan, X.; Jiao, J.; Chen, N.; Huang, F.; Di, L.; Wang, J.; Sha, Z.; Liu, J. Geoscience model service integrated workflow for rainstorm waterlogging analysis. *Int. J. Digit. Earth.* **2021**, *14*(7), 851–873, doi:10.1080/17538947.2021.1898686
37. Bychkov, I.; Feoktistov, A.; Gorsky, S.; Kostromin, R. Agent-based Intellectualization of Continuous Software Integration and Resource Configuration for Distributed Computing. In Proceedings of the 7th International Conference on Information Technology and Nanotechnology, Samara, Russia, 20–24 September 2021, doi:10.1109/ITNT52450.2021.9649396
38. Edelev, A.; Gorsky, S.; Feoktistov, A.; Bychkov, I.; Marchenko, M. Development of means of distributed computing management in Orlando Tools. In Proceedings of the 17th International Conference on Parallel Computational Technologies, St. Petersburg, Russia, 28–30 March, 2023. Available online: https://www.elibrary.ru/download/elibrary_53852349_80610177.pdf (accessed on 22 September 2023) (In Russian)
39. Irkutsk Supercomputer Center. Available online: <https://hpc.icc.ru/> (accessed on 22 September 2023)
40. Bychkov, I.V.; Ruzhnikov, G.M.; Fedorov, R.K.; Khmel'nov, A.E.; Popova, A.K. Organization of digital monitoring of the Baikal natural territory. *IOP C. Ser. Earth Env.* **2021**, *629*(1), 012067, doi:10.1088/1755-1315/629/1/012067
41. Johansson, J.; Hassel, H. Modelling, simulation and vulnerability analysis of interdependent technical infrastructures. In Risk and Interdependencies in Critical Infrastructures: A Guideline for Analysis. Hokstad, P., Utne, I.B., Vatn, J., Eds.; Springer-Verlag: London, UK, 2012; pp. 49–66, doi:10.1007/978-1-4471-4661-2_5
42. Meng, F.; Fu, G.; Farmani, R.; Sweetapple, C.; Butler, D. Topological attributes of network resilience: A study in water distribution systems. *Water Res.* **2018**, *143*, 376–386, doi:10.1016/j.watres.2018.06.048
43. Alderson, D.L.; Brown, G.G.; Carlyle, W.M. Assessing and improving operational resilience of critical infrastructures and other systems. In Tutorials in Operations Research: Bridging Data and Decisions. Newman, A, Leung, J, Eds.; INFORMS: Catonsville, MD, 2014, pp. 180–215, doi:10.1287/educ.2014.0131
44. Wang, W.; Zhang, Y.; Li, Y.; Liu, C.; Han, S. Vulnerability analysis of a natural gas pipeline network based on network flow. *International Journal of Pressure Vessels and Piping*, 2020, *188*, 104236, doi:10.1016/j.ijpvp.2020.104236
45. Su, H.; Zio, E.; Zhang, J.; Li, X. 2018. A systematic framework of vulnerability analysis of a natural gas pipeline network. *Reliab. Eng. Syst. Safe.* **2018**, *175*, 79–91, doi:10.1016/j.res.2018.03.006
46. Dwivedi, A.; Yu, X. A maximum-flow-based complex network approach for power system vulnerability analysis. *IEEE T. Ind. Inform.* **2011**, *9*(1), 81–88, doi:10.1109/TII.2011.2173944
47. Xie, B.; Tian, X.; Kong, L.; Chen, W. The vulnerability of the power grid structure: A system analysis based on complex network theory. *Sensors* **2021**, *21*(21), 7097, doi 10.3390/s21217097
48. Bychkov, I.V.; Gorsky, S.A.; Edelev, A.V.; Kostromin, R.O.; Sidorov, I.A.; Feoktistov, A.G.; Fereferov, E.S.; Fedorov, R.K. Support for Managing the Survivability of Energy Systems Based on a Combinatorial Approach. *J. Comput. Sys. Sc. Int.* **2021**, *60*(6), 981–994, doi:10.1134/S1064230721060071

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.