Программные системы и комплексы

УДК 004.75
DOI:10.25729/ESI.2025.39.3.013

Использование технологии In-Memory Data Grid при исследовании живучести энергетических инфраструктур

Воскобойников Михаил Леонтьевич, Феоктистов Александр Геннадьевич

Институт динамики систем и теории управления им. В.М. Матросова СО РАН,
Россия, Иркутск, *mikev1988@icc.ru*

Аннотация. Исследование и повышение живучести энергетических инфраструктур является актуальной проблемой, сопряженной с высокой вычислительной сложностью подготовки и проведения экспериментов. Сложность экспериментов обусловливается необходимостью учета больших наборов сценариев крупных внешних возмущений, воздействующих на исследуемую инфраструктуру, выявления ее критических элементов, отказ которых может привести к существенным сбоям в генерации, транспортировке и поставке энергоресурсов, а также в планировании мероприятий, направленных на повышение живучести данной инфраструктуры. Решение подобных задач в вычислительной среде на основе выполнения научных рабочих процессов, использующих распределенную базу данных в оперативной памяти узлов среды, позволяет существенно сократить время расчетов. Однако такой подход не поддерживается в известных системах управления рабочими процессами. В статье предложен новый подход к реализации анализа живучести энергетических инфраструктур с помощью инструментального комплекса Framework for Development and Execution of Scientific WorkFlows с использованием распределенных баз данных. В частности, создан научный рабочий процесс для анализа живучести энергетических инфраструктур, разработана методика прогнозирования требуемого размера оперативной памяти для его выполнения, реализован набор испытательных стендов (системных рабочих процессов) для выполнения расчетов по данной методике с учетом ключевых параметров предметной области, оказывающих существенное влияние на изменение размера данных. Проведен вычислительный эксперимент, демонстрирующий точность прогнозирования требуемого размера оперативной памяти при исследовании двух тестовых моделей энергетических инфраструктур различной сложности.

Ключевые слова: энергетическая инфраструктура, моделирование, распределенная вычислительная среда, вычисления в оперативной памяти, рабочие процессы, автоматизация

Цитирование: Воскобойников М.Л. Использование технологии In-Memory Data Grid при исследовании живучести энергетических инфраструктур / М.Л. Воскобойников, А.Г. Феоктистов // Информационные и математические технологии в науке и управлении, 2025. – № 3(39) – С. 154-163. – DOI:10.25729/ESI.2025.39.3.013.

Введение. В настоящее время исследование и повышение живучести энергетических инфраструктур является актуальной проблемой [1]. Решение этой проблемы, как правило, сопряжено с высокой вычислительной сложностью подготовки и проведения экспериментов. Сложность экспериментов обусловливается необходимостью учета множества сценариев крупных внешних возмущений, воздействующих на инфраструктуру, выявления ее критических элементов, отказ которых может привести к существенным сбоям в генерации, транспортировке и поставке энергоресурсов, а также в планировании мероприятий, направленных на повышение живучести исследуемой инфраструктуры.

В статье предложен новый подход к автоматизации подготовки и проведения крупномасштабных экспериментов по исследованию и повышению живучести энергетических инфраструктур в распределенной вычислительной среде (РВС) на основе разработки и выполнения научных рабочих процессов (НРП). Ключевая особенность подхода заключается в использовании технологии In-Memory Data Grid (IMDG), обеспечивающей возможность хранения и обработки расчетных данных в оперативной памяти (ОП) узлов РВС [2]. Применение IMDG позволяет существенно сократить время вычислений.

Создание НРП и управление ими в РВС осуществляется с помощью инструментального комплекса Framework for Development and Execution of Scientific WorkFlows (FDE-SWFs) [3].

FDE-SWFs относится к классу систем управления рабочими процессами [4]. В отличие от известных систем этого класса он поддерживает автоматизацию развертывания, хранения и обработки расчетных данных в ОП узлов среды. В рамках FDE-SWFs реализована новая методика прогнозирования требуемого размера ОП и выделения необходимого числа узлов кластера IMDG на основе знаний о структуре исходной базы данных (БД) и ключевых параметрах предметной области, влияющих на изменение размера данных.

Средства поддержки IMDG. Системы хранения данных на основе IMDG (In-Memory Data Grid) имеют неоспоримое преимущество в скорости обработки данных по сравнению с традиционными БД [5]. Кроме того, имеется большой спектр инструментов для поддержки технологии IMDG на ресурсах высокопроизводительных вычислений [6, 7]. Каждый инструмент представляет собой связующее программное обеспечение для поддержки распределенной обработки больших наборов данных. Hazelcast [8], Infinispan [9] и Apache Ignite [10] являются наиболее популярным свободно распространяемым связующим ПО на основе IMDG. Указанные программные системы имеют схожую функциональность и производительность. В частности, они поддерживают клиентские API на языках Java, C++ и Python. Конечные пользователи могут использовать их для создания кластеров IMDG с распределенными кэшами «ключ/значение». Однако каждая из этих программных систем может иметь свои преимущества в сравнении с другими системами относительно определенного набора данных и конкретного сценария их обработки.

Hazelcast поддерживает динамические операции с данными в режиме реального времени. Эта система объединяет высокопроизводительную потоковую обработку с быстрым хранением данных. Некоторые компоненты Hazelcast распространяются в соответствии с Лицензионным соглашением сообщества Hazelcast версии 1.0.

Infinispan обеспечивает гибкие возможности развертывания и надежные средства хранения, управления и обработки данных. Эта система поддерживает и распределяет данные типа «ключ/значение» на масштабируемых кластерах IMDG с высокой доступностью и отказоустойчивостью. Infinispan доступна по лицензии Apache 2.0.

С помощью Apache Ignite можно создавать полнофункциональные, децентрализованные транзакционные БД типа «ключ/значение» с удобным и простым в использовании интерфейсом для работы с данными большого размера в режиме реального времени, включая асинхронные вычисления в ОП. Apache Ignite поддерживает архитектуру долговременной памяти с ускорителем больших данных в памяти и на диске для данных, вычислений, сервисов и потоковых сетей. Эта система предоставляет открытый исходный код, распространяемый под лицензией Apache License 2.0. В отличие от Hazelcast, полная функциональность Apache Ignite и Infinispan бесплатна.

К преимуществам Apache Ignite относится поддержка горизонтально масштабируемой и отказоустойчивой распределенной БД SQL, которая полностью соответствует стандарту ANSI-99. Hazelcast и Infinispan поддерживают аналогичные SQL-запросы с исключениями. Apache Ignite допускает развертывания своего ПО динамически во время выполнения рабочих процессов приложения. Для Hazelcast и Infinispan это ключевое требование является трудоемким процессом. Как правило, конечные пользователи Hazelcast и Infinispan сначала создают кластер IMDG и только потом запускают задачи обработки данных на узлах развернутого кластера [11-13]. Кроме того, разработчиками Apache Ignite реализована методика определения числа узлов кластера IMDG на основе прогнозируемого размера ОП, необходимого для обработки данных [14]. Аналогичная методика была разработана и для Infinispan. Тем не менее, Apache Ignite дополнительно учитывает накладные расходы памяти при хранении данных и использование диска. Hazelcast предоставляет только примеры, которые можно экстраполировать для конкретных рабочих нагрузок.

Методика прогнозирования требуемого объема ОП. Для прогнозирования требуемого размера ОП S при решении задач с использованием кластера Apache Ignite разработана методика, базирующаяся на расчетах по формулам (1)-(3):

$$S = D + (o_{data} + o_{index}) \sum_{i=1}^{t} r_i(x_i, x_2, \dots, x_h), \quad (1)$$

$$D = \sum_{i=1}^{t} d_i, \quad (2)$$

$$d_i = r_i(x_1, x_2, \dots, x_h) \sum_{j=1}^{c_i} s_{ij}, \quad (3)$$

где переменные интерпретируются следующим образом:

- $D > 0$ – суммарный размер данных в байтах во всех таблицах;
- $t > 0$ – число таблиц;
- $d_i > 0$ – размер данных в i-ой таблице в байтах, рассчитываемый на основе информации о структуре данных;
- $o_{data} > 0$ – удельные накладные расходы на хранение данных, рассчитываемые для числа записей не меньше 16000000;
- $o_{index} > 0$ – удельные накладные расходы на хранение индексов, рассчитываемые для числа записей не меньше 16000000;
- $c_i > 0$ – число столбцов в i-ой таблицы;
- $r_i(x_1, x_2, \dots, x_h) > 0$ – число строк в i-ой таблице (функция r_i конкретизируется для каждой i-й таблицы с учетом ключевых параметров предметной области);
- $x_1, x_2, \dots, x_h$ – ключевые параметры предметной области, влияющие на изменение размера обрабатываемых данных;
- $h > 0$ – число ключевых параметров;
- $s_{ij} > 0$ – максимальный размер данных в j-ом столбце i-ой таблицы в байтах.

Значения o_{data} и o_{index} рассчитываются для каждой задачи предметной области при однократном занесении в распределенную БД не менее 16000000 записей. При достижении такого числа записей эти значения стабилизируется.

Расчет накладных расходов для хранения данных выполняется по формуле:

$$o_{data} = \frac{M_{data} - D}{\sum_{i=1}^{t} r_i(x_1, x_2, \dots, x_h)}, \quad (4)$$

где M_{data} – это размер данных без учета индексов таблиц, который занимает БД, размещенная на вычислительных узлах.

Расчет накладных расходов для хранения индексов выполняется по формуле:

$$o_{index} = \frac{M_{index} - M_{data}}{\sum_{i=1}^{t} r_i(x_1, x_2, \dots, x_h)}, \quad (5)$$

где M_{index} – это размер данных с учетом индексов таблиц, который занимает БД, размещенная на вычислительных узлах.

Расчет накладных расходов на данные проводится с помощью следующего алгоритма:

I. Модификация исходной БД, в которой отключаются индексы во всех таблицах.
II. Занесение в БД 16000000 или более записей.
III. Выполнение следующих двух запросов к Apache Ignite с помощью SQL-консоли, которая входит в его состав:
   а. Получение размера БД в ОП в байтах.
   б. Получение числа записей в БД во всех таблицах.
IV. Вычисление удельных накладных расходов на хранение одной записи в БД.

Расчет накладных расходов на индексы проводится с помощью следующего алгоритма:

I. Модификация исходной БД, в которой включаются индексы во всех таблицах.
II. Занесение в БД 16000000 или более записей.

III. Выполнение следующих двух запросов к Apache Ignite с помощью SQL-консоли, которая входит в его состав:
   а. Получение размера БД в ОП в байтах.
   б. Получение числа записей в БД во всех таблицах.

IV. Вычисление удельных накладных расходов на хранение индексов.

Функция $r_i(x_1, x_2, \dots, x_h)$ конкретизируется для каждой i-ой таблицы с учетом ключевых параметров предметной области $x_1, x_2, \dots, x_h$. Предложенная методика является универсальной для прогнозирования требуемого размера ОП при решении задач с использованием кластеров Apache Ignite в разных предметных областях. Специфика предметной области определяется ее ключевыми параметрами $x_1, x_2, \dots, x_h$. В отличие от методик, представленных в [14, 15], данная методика не использует субъективные оценки размера ОП, требуемой для хранения данных и индексов. Она позволяет достаточно точно рассчитать требуемый объем ОП, опираясь на знания о структуре исходных данных и ключевых параметрах предметной области, а также на результаты одного конкретного эксперимента. Число узлов, выделяемых в дальнейшем для решения задачи, определяется следующей формулой:

$$1 \le y = \left\lceil \frac{s}{p_l(1-k_l^{os})} \right\rceil \le q, \tag{6}$$

где p_l – доступный объем памяти l-го узла в ГБ, $0 < k_l^{os} \le 1$ – коэффициент использования ОП операционной системой l-го узла, q – квота на число узлов РВС для решения задачи пользователем, определяемая административной политикой РВС, $l \in \overline{1,e}$, e – число узлов РВС. Предполагаем, что все узлы РВС являются однородными.

Вычислительный эксперимент. Пусть конфигурация энергетической инфраструктуры описывается сетью $G = (V, U)$, где V – упорядоченное множество узлов, U – упорядоченное множество дуг. Сеть G является направленным графом, поэтому для каждой дуги $(i, j) \in U$ узел $i \in V$ является началом, а узел $j \in V$ – концом. Известные исследования по анализу уязвимости [16], как правило, основаны на оценке последствий множества сценариев возмущений, моделирующих отказ групп из k элементов энергетической инфраструктуры. Множеством отказов F размером $k \ge 1$ называется группа из k номеров элементов сети G, отказ которых наступает одновременно: $F = \{c_l : 1 \le c_l \le |V| + |U|,\ l = (1, k)\}$, где c_l – это номер элемента сети G. Максимальное число возмущений рассчитывается по формуле (1):

$$f(n, m, k) = \sum_{i=1}^{k} \frac{(n+m)!}{((n+m-i)!\, i!)}, \tag{7}$$

где m и n – это число дуг и узлов соответственно, которые задаются экспертом. Переменные m, n и k являются ключевыми параметрами предметной области, влияющими на изменение размера обрабатываемых данных.

Структура исходных данных включает 6 таблиц: DISTURBANCE, MES_ELEMENTS, SOLUTION_REPORTS, FAILED_ELEMENTS, TRANSPORT_DATA и REGIONAL_DATA. На рис. 1 представлен НРП для полного перебора множеств отказов размера k. Он включает следующие операции: заполнение распределенной БД данными о последствиях возмущений (o_1); обработка распределенной БД средствами Apache Ignite для оценки критичности элементов сети (o_2). В операциях используются следующие параметры: z_1 – БД с конфигурацией исследуемой энергетической инфраструктуры; z_2 – результат запуска Apache Ignite; z_3 – размер группы отказов; z_4 – число сценариев возмущений; z_5 – адреса узлов кластера Apache Ignite; z_6 – результат заполнения распределенной БД; z_7 – результат оценки критичности элементов сети.

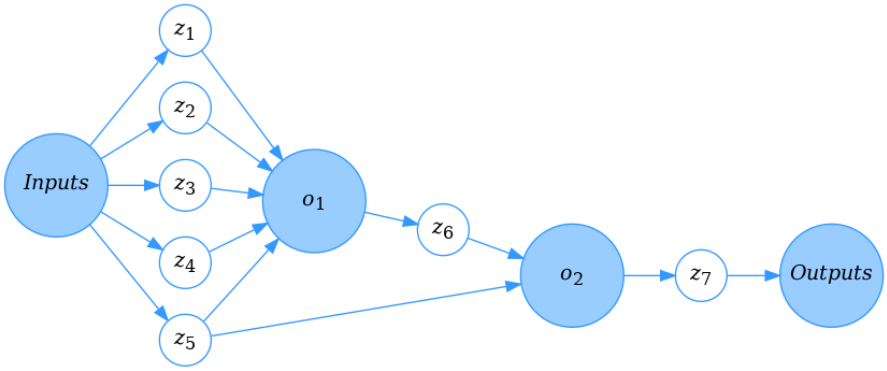

Рис. 1. НРП

Исходная БД включает таблицы DISTURBANCE, MES_ELEMENTS, SOLUTION_REPORTS, FAILED_ELEMENTS, TRANSPORT_DATA и REGIONAL_DATA, для каждой из которых определены соответственно функции $r_1 - r_6$. В рамках предложенной методики размеры этих таблиц рассчитываются соответственно по формулам (8)-(10), (11)-(13), (14)-(16), (17)-(19), (20)-(22) и (23)-(25), полученным путем конкретизации (2)-(3) с учетом ключевых параметров предметной области:

$$d_1 = r_1(n,m,k)\sum_{j=1}^{c_1} s_{1j}, \quad (8)$$

$$r_1(n,m,k) = \sum_{i=1}^{k} \frac{(n+m)!}{((n+m-i)!i!)}, \quad (9)$$

$$s_1 = \sum_{j=1}^{c_1} s_{1j}, \quad (10)$$

$$d_2 = r_2(n,m)\sum_{j=1}^{c_2} s_{2j}, \quad (11)$$

$$r_2(n,m) = n + m, \quad (12)$$

$$s_2 = \sum_{j=1}^{c_2} s_{2j}, \quad (13)$$

$$d_3 = r_3(n,m,k)\sum_{j=1}^{c_3} s_{3j}, \quad (14)$$

$$r_3(n,m,k) = r_1(n,m,k), \quad (15)$$

$$s_3 = \sum_{j=1}^{c_3} s_{3j}, \quad (16)$$

$$d_4 = r_4(n,m,k)\sum_{j=1}^{c_4} s_{4j}, \quad (17)$$

$$r_4(n,m,k) = \sum_{i=1}^{k}[ir_1(n,m,i)], \quad (18)$$

$$s_4 = \sum_{j=1}^{c_4} s_{4j}, \quad (19)$$

$$d_5 = r_5(n,m,k)\sum_{j=1}^{c_5} s_{5j}, \quad (20)$$

$$r_5(n,m,k) = r_1(n,m,k)|U|, \quad (21)$$

$$s_5 = \sum_{j=1}^{c_5} s_{5j}, \quad (22)$$

$$d_6 = r_6(n,m,k)\sum_{j=1}^{c_6} s_{6j}, \quad (23)$$

$$r_6(n,m,k) = r_1(n,m,k)|V|, \quad (24)$$

$$s_6 = \sum_{j=1}^{c_6} s_{6j}. \quad (25)$$

В качестве примера энергетических инфраструктур рассмотрены тестовые модели системы газоснабжения, состоящей из 26 дуг и 20 узлов ($k = 8, m = 26$ и $n = 0$), и топливно-энергетического комплекса (ТЭК), включающего 2190 дуг и 1220 узлов ($k = 2, m = 2190, n = 1220$). По формулам (1)-(5) и (8)-(25) спрогнозирован требуемый размер ОП для исследования живучести указанных инфраструктур с помощью НРП на кластере Apache Ignite для двух пулов ресурсов: 8 узлов со следующими характеристиками – AMD Ryzen 9 5900X 12-Core Processor, 128 ГБ ОЗУ (пул 1); 2 узла со следующими характеристиками – AMD EPYC 9654 96-Core Processor, 768 ГБ ОЗУ (пул 2). Для модели системы газоснабжения экспериментальным путем получено, что величины накладных расходов на хранение данных

(o_{data}) и индексов (o_{index}) стабилизируется на значениях 240 байт и 172 байта соответственно – рис. 2 (а) и 2 (б). Для модели ТЭК величины накладных расходов на хранение данных (o_{data}) и индексов (o_{index}) стабилизируются на значениях 20 байт и 206 байт соответственно – рис. 2 (в) и 2 (г).

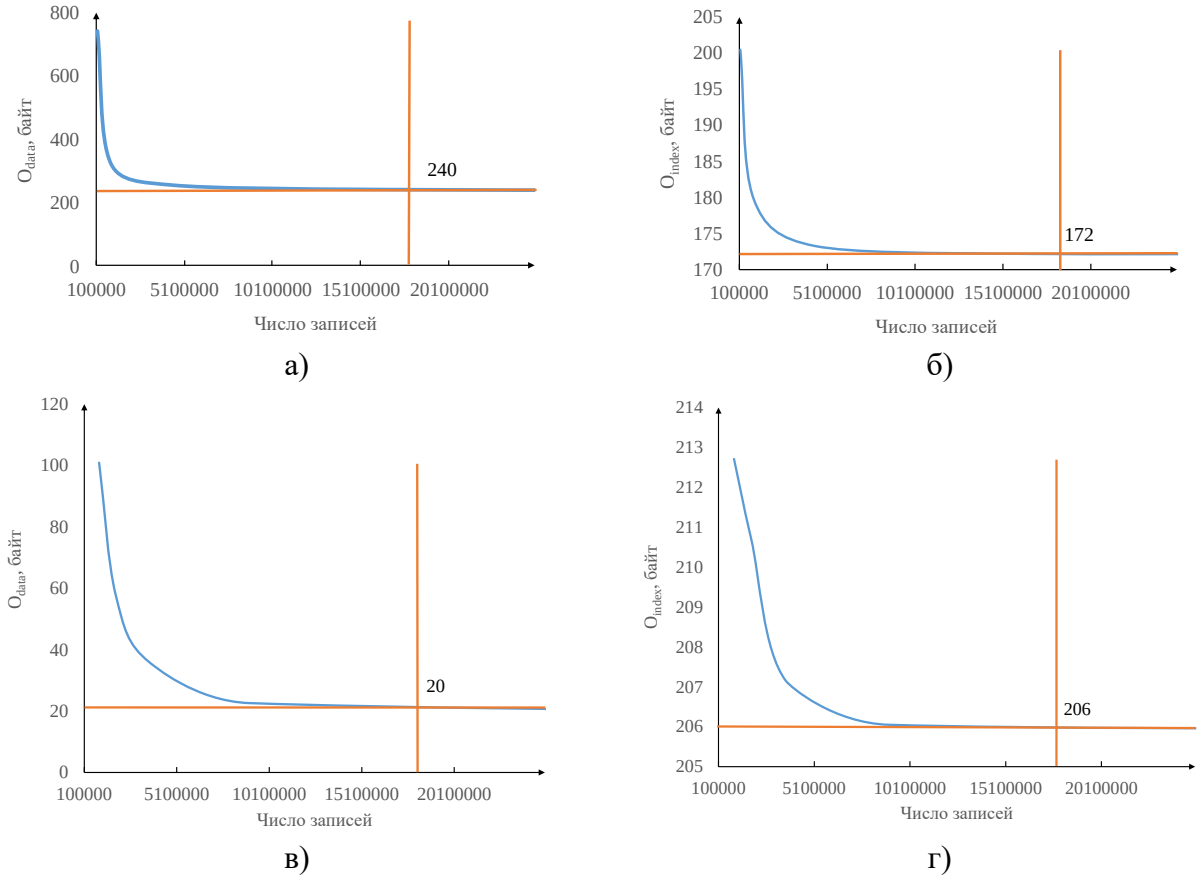

Рис. 2. Удельные расходы на хранение данных (а) и индексов (б)

Методика (М), значения ключевых параметров (k, m и n), число записей (ЧЗ), пул используемых вычислительных ресурсов, прогнозируемый размер ОП (ПР ОП), фактический размер ОП (ФР ОП), абсолютная погрешность прогноза (АПП) в сравнении с ФР ОП, относительная погрешность прогноза (ОПП), прогнозируемое число узлов (ПЧУ) и фактическое число узлов (ФЧУ) для двух моделей разной сложности приведены в таблице 1.

Таблица 1. Результаты экспериментов

М	k / m / n	Пул	ЧЗ	ПР ОП, ГБ	ФР ОП, ГБ	АПП, ГБ	ОПП, %	ПЧУ	ФЧУ
M1	8 / 26 / 0	1	140512730	62,837	61,887	0,950	1,54	1	1
M2				119,036		57,149	92,34	2	
M3				11,357		-50,530	81,65	1	
M1	8 / 26 / 0	2	140512730	62,837	61,887	0,950	1,54	1	1
M2				119,036		57,149	92,34	1	
M3				11,357		-50,530	81,65	1	
M1	2 / 2190 / 1220	1	902411450	479,012	473,214	5,798	1,23	6	6
M2				621,846		148,632	31,41	7	
M3				376,964		-96,25	20,34	5	
M1	2 / 2190 / 1220	2	902411450	479,012	473,214	5,798	1,23	1	1
M2				621,846		148,632	31,41	2	
M3				376,964		-96,25	20,34	1	

Проведено сравнение трех методик прогнозирования требуемого размера ОП: предложенная методика М1, методика М2, представленная в [15], и методика М3

разработчиков Apache Ignite. Исходя из результатов экспериментов, очевидно, что методика М1 обеспечивает наименьшую ОПП, не превышающую 1,54% на всех тестах для обеих моделей энергетических инфраструктур разной сложности, в сравнении с методиками М2 и М3, показывающими ОПП не менее 31,41% и 20,34% соответственно. При этом М1 обеспечивает отклонение ПР ОП только в большую сторону. Методика М2 дает более завышенный прогноз, что может снижать эффективность использования вычислительных ресурсов. В тоже время методика М3 прогнозирует недостаточный размер ОП, что зачастую приводит к выделению недостаточного числа узлов и отказу вычислительного процесса без использования дополнительной виртуальной памяти на диске или к существенному увеличению времени вычислений более чем в 10 раз при использовании дисковой памяти. В частности, для решения задачи по модели ТЭК на пуле 1 по методике М3 прогнозируется недостаточное число узлов.

Для расчета накладных расходов на хранение данных и индексов в рамках методики М1 разработаны испытательные стенды в виде системных рабочих процессов (СРП) СРП1 и СРП2, представленных соответственно на рис. 3 и 4. Системные операции СРП выделены красным цветом. СРП1 включает следующие операции: запуск кластера Apache Ignite для создания БД (o_3); запуск НРП (o_4); расчет расходов на хранение данных (o_5); остановка кластера Apache Ignite (o_6). В операциях используются следующие параметры: $z_1 - z_5$, z_7 – параметры НРП, представленного на рис. 1; z_8 – конфигурационный файл Apache Ignite, описывающий структуру БД с отключенными индексами для всех таблиц; z_9 – файл с результатами расчета накладных расходов на хранение данных; z_{10} – результат остановки кластера Apache Ignite.

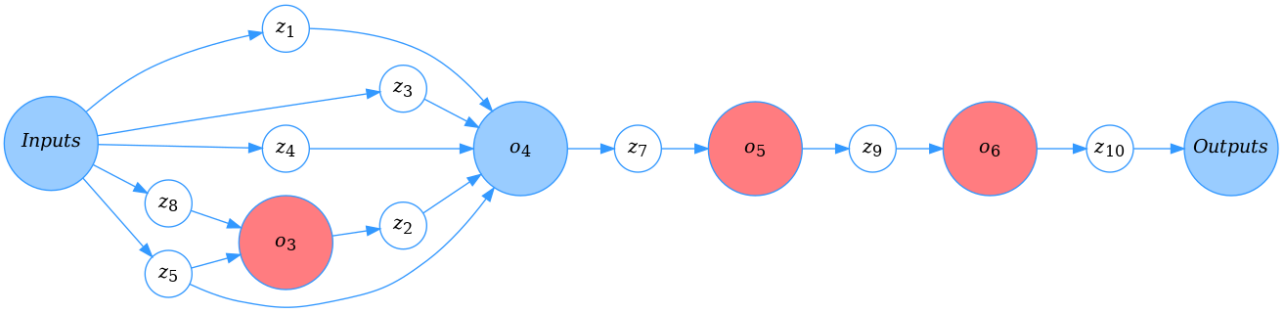

Рис. 3. СРП1

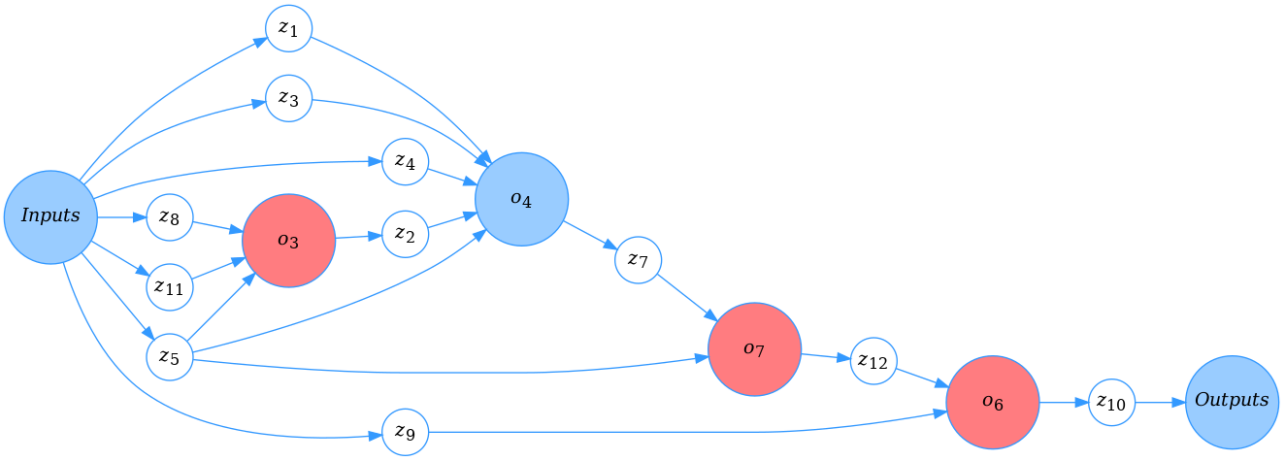

Рис. 4. СРП2

СРП2 включает следующие операции: o_3, o_4 и o_6 – операции, описанные выше; расчет расходов на хранение индексов (o_7). В операциях используются следующие параметры: $z_1 - z_5$ и $z_7 - z_{10}$ – параметры, описанные выше; z_{11} – конфигурационный файл Apache Ignite,

описывающий структуру БД с включенными индексами для всех таблиц БД; z_{12} – файл с результатами расчета накладных расходов на хранение индексов.

Для прогнозирования требуемого размера ОП и формирования конфигурационного файла кластера Apache Ignite разработан СРПЗ (рис. 5). СРПЗ включает следующие операции: расчет требуемого размера ОП (o_8); модификация шаблона конфигурационного файла (o_9). В операциях используются следующие параметры: z_1, z_2, z_9 и z_{11} – параметры, описанные выше; z_{14} – шаблон конфигурационного файла; z_{15} – размер требуемого размера ОП; z_{16} – конфигурационный файл.

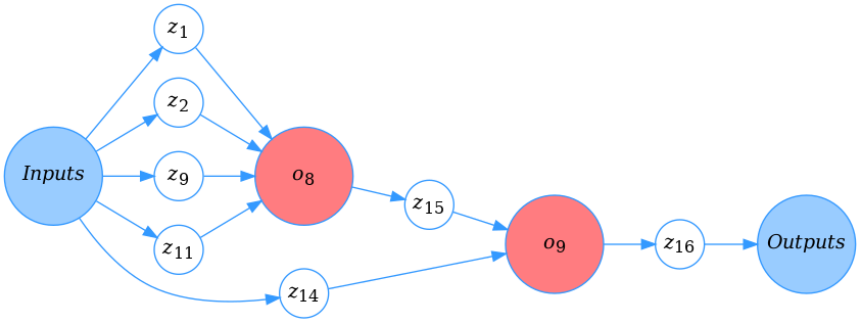

Рис 5. СРПЗ

Заключение. В статье предложен новый подход к автоматизации подготовки и проведения крупномасштабных экспериментов при решении задачи исследования и повышения живучести энергетических инфраструктур в РВС с использованием IMDG. Приведен пример организации кластера IMDG, использованного для решения задачи. Получены оценки накладных расходов на хранение данных и индексов, которые позволили с высокой точностью спрогнозировать требуемый размер ОП для размещения распределенной БД на вычислительных узлах.

Благодарности. Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации, проект № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9).

Список источников

1. Jasiūnas J., Lund P. D., Mikkola J. Energy system resilience–A review. Renewable and sustainable energy reviews, 2021, vol. 150, pp. 111476, DOI: 10.1016/j.rser.2021.111476.
2. Mukuhi D.K., Outtagarts A. Distributed Intelligent Resource Orchestration. 2023 2nd International Conference on 6G Networking (6GNet), Paris, France, 2023, pp. 1-3, DOI: 10.1109/6GNet58894.2023.10317732.
3. Voskoboinikov M., Feoktistov A., Tchernykh A. Framework for development and execution of scientific WorkFlows: Designing service-oriented applications. Programming and computer software, 2024, vol. 50, no. 8, pp. 900-913, DOI: 10.1134/S0361768824700828.
4. Воскобойников М.Л. Сравнительный анализ систем управления научными рабочими процессами / М.Л. Воскобойников, А.Г. Феоктистов // Информационные и математические технологии в науке и управлении, 2024. – № 3(35). – С. 102-111. – DOI: 10.25729/ESI.2024.35.3.009.
5. 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, vol. 11, pp. 65, DOI: 10.3390/computation11030065.
6. Grover P., Kar A.K., Big data analytics: A review on theoretical contributions and tools used in literature. Glob. J. Flex. Syst.Manag. 2017, vol. 18, pp. 203-229, DOI: 10.1007/s40171-017-0159-3.
7. de Souza Cimino L., de Resende J.E.E., Silva L.H.M., et al. A middleware solution for integrating and exploring IoT and HPC capabilities. Software: Practice and experience. 2019, vol. 49, pp. 584–616. DOI: 10.1002/spe.2630.
8. Hazelcast, available at: https://hazelcast.com (accessed: 04/29/2025).
9. Infinispan. In-Memory distributed data store, available at: https://infinispan.org (accessed: 04/29/2025).
10. Apache Ignite. Distributed database for high-performance applications with in-memory speed, available at: https://ignite.apache.org (accessed: 04/29/2025).

11. 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, UK, 8–11 December 2014, pp. 79-88, DOI: 10.1109/UCC.2014.16.
12. Zhou M., Feng D. Application of in-memory computing to online power grid analysis. IFAC-PapersOnLine, 2018, vol. 51, iss. 28, pp.132-137, DOI: 10.1016/j.ifacol.2018.11.690.
13. Zhou M., Yan J., Wu Q. Graph computing and its application in power grid analysis. CSEE J. Power energy syst. 2022, vol. 8, iss. 6, pp. 1550-1557, DOI: 10.17775/CSEEJPES.2021.00430.
14. Capacity Planning, available at: https://apacheignite.readme.io/docs/capacity-planning (accessed: 04/29/2025).
15. Feoktistov A.G., Edelev A.V., Tchernykh A.N., et al. An approach to implementing high-performance computing for problem solving in Workflow-Based energy infrastructure resilience studies. Computation, 2023, vol. 11, iss. 12, 243 p, DOI: 10.3390/computation11120243.
16. Jonsson H., Johansson J., Johansson H. Identifying critical components in technical infrastructure networks. Proceedings of the Institution of Mechanical Engineers. Part O, 2008, vol. 222, iss. 2, pp. 235-243, DOI: 10.1243/1748006XJRR138.

Воскобойников Михаил Леонтьевич. *Младший научный сотрудник Института динамики систем и теории управления им. В.М. Мелентьева СО РАН. Область научных интересов – программная инженерия, распределенные вычисления, сервис-ориентированный подход, испытательные стенды. AuthorID: 1102663, SPIN: 3417-0258, ORCID: 0000-0003-3034-4907. mikev1988@icc.ru, Россия, г. Иркутск, Лермонтова, 134.*

Феоктистов Александр Геннадьевич. *Д.т.н., доцент, главный научный сотрудник Института динамики систем и теории управления им. В.М. Матросова СО РАН. Область научных интересов – распределенные вычисления, системы управления рабочими процессами, автоматизация разработки научных приложений, концептуальное моделирование. AuthorID: 1535, SPIN: 5743-1777, ORCID: 0000-0002-9127-6162, agf@icc.ru, Россия, г. Иркутск, Лермонтова,134.*

UDC 004.75
DOI:10.25729/ESI.2025.39.3.013

Using In-Memory Data Grid technology in energy infrastructure resilience analysis

Mikhail L. Voskoboinikov, Alexander G. Feoktistov

Matrosov Institute for system dynamics and control theory of SB RAS,
Russia, Irkutsk, *mikev1988@icc.ru*

Abstract. Studying and enhancing the resilience of energy infrastructure is an actual problem that is associated with high computational complexity of preparing and conducting experiments. The complexity of experiments is due to the range of important factors. These factors include large sets of scenarios for significant external disturbances affecting the infrastructure under study, identification of its critical elements whose failure could lead to significant disruptions in the energy resource generation, transportation, and supply, as well as planning activities aimed at enhancing this infrastructure's resilience. Solving these problems in a computing environment based on executing scientific workflows using a distributed database in the RAM of the environment nodes can significantly reduce the computation time. However, this approach is not supported by known workflow management systems. In this context, we propose a new approach to implement the analyzing the energy infrastructure resilience using the Framework for Development and Execution of Scientific WorkFlows and distributed databases. Specifically, we created a scientific workflow to analyze energy infrastructure resilience. Next, we developed a method for predicting the required RAM size for workflow execution. Then, we implemented a set of testbeds (system workflows) to perform computing according to this method. This method takes into account key parameters in the subject area that significantly impact changes in data size. Finally, we conducted a computational experiment to demonstrate the accuracy of predicting the required RAM size when studying two test models of energy infrastructures of different complexities.

Keywords: energy infrastructure, modeling, distributed computing environment, in-memory computing, workflows, automation

Acknowledgements: The study was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. FWEW-2021-0005 “Technologies for the development and analysis of subject-oriented intelligent group control systems in non-deterministic distributed environments” (reg. no. 121032400051-9).

References

1. Jasiūnas J., Lund P. D., Mikkola J. Energy system resilience–A review. Renewable and sustainable energy reviews, 2021, vol. 150, pp. 111476, DOI: 10.1016/j.rser.2021.111476.
2. Mukuhi D. K., Outtagarts A. Distributed Intelligent Resource Orchestration. 2023 2nd International Conference on 6G Networking (6GNet), Paris, France, 2023, pp. 1-3, DOI: 10.1109/6GNet58894.2023.10317732.
3. Voskoboinikov M., Feoktistov A., Tchernykh A. Framework for development and execution of scientific WorkFlows: Designing service-oriented applications. Programming and computer software, 2024, vol. 50, no. 8, pp. 900-913,. DOI: 10.1134/S0361768824700828.
4. Voskoboynikov M.L., Feoktistov A.G. Sravnitel'nyy analiz sistem upravleniya nauchnymi rabochimi protsessami [Comparative analysis of scientific workflow management systems]. Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii [Information and mathematical technologies in science and management], 2024, no. 3(35), pp. 102-111, DOI: 10.25729/ESI.2024.35.3.009.
5. 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, vol. 11, pp. 65, DOI: 10.3390/computation11030065.
6. Grover P., Kar A.K., Big data analytics: A review on theoretical contributions and tools used in literature. Glob. J. Flex. Syst.Manag. 2017, vol. 18, pp. 203-229, DOI: 10.1007/s40171-017-0159-3.
7. de Souza Cimino L., de Resende J.E.E., Silva L.H.M., et al. A middleware solution for integrating and exploring IoT and HPC capabilities. Software: Practice and experience. 2019, vol. 49, pp. 584–616. DOI: 10.1002/spe.2630.
8. Hazelcast, available at: https://hazelcast.com (accessed: 04/29/2025).
9. Infinispan. In-Memory distributed data store, available at: https://infinispan.org (accessed: 04/29/2025).
10. Apache Ignite. Distributed database for high-performance applications with in-memory speed, available at: https://ignite.apache.org (accessed: 04/29/2025).
11. 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, UK, 8–11 December 2014, pp. 79-88, DOI: 10.1109/UCC.2014.16.
12. Zhou M., Feng D. Application of in-memory computing to online power grid analysis. IFAC-PapersOnLine, 2018, vol. 51, iss. 28, pp.132-137, DOI: 10.1016/j.ifacol.2018.11.690.
13. Zhou M., Yan J., Wu Q. Graph computing and its application in power grid analysis. CSEE J. Power energy syst. 2022, vol. 8, iss. 6, pp. 1550-1557, DOI: 10.17775/CSEEJPES.2021.00430.
14. Capacity Planning, available at: https://apacheignite.readme.io/docs/capacity-planning (accessed: 04/29/2025).
15. Feoktistov A.G., Edelev A.V., Tchernykh A.N., et al. An approach to implementing high-performance computing for problem solving in Workflow-Based energy infrastructure resilience studies. Computation, 2023, vol. 11, iss. 12, 243 p, DOI: 10.3390/computation11120243.
16. Jonsson H., Johansson J., Johansson H. Identifying critical components in technical infrastructure networks. Proceedings of the institution of mechanical engineers. Part O, 2008, vol. 222, iss. 2, pp. 235-243, DOI: 10.1243/1748006XJRR138.

Voskoboinikov Mikhail Leontevich. *Junior Researcher at the Matrosov institute for system dynamics and control theory of SB RAS. The main direction of research – program engineering, distributed computing, workflow management systems, service-oriented paradigm, testbeds. AuthorID: 1102663, SPIN: 3417-0258, ORCID: 0000-0003-3034-4907, mikev1988@icc.ru, Russia, Irkutsk, Lermontova, 134*.

Feoktistov Alexander Gennadevich. *Doctor of technical sciences, associate professor, chief researcher at the Matrosov institute for system dynamics and control theory of SB RAS. The main direction of research – distributed computing, workflow management systems, automation of developing scientific applications, conceptual modeling. AuthorID: 1535, SPIN: 5743-1777, ORCID: 0000-0002-9127-6162, agf@icc.ru, Russia, Irkutsk, Lermontova, 134.*

Статья поступила в редакцию 13.05.2025; одобрена после рецензирования 01.06.2025; принята к публикации 26.08.2025.

The article was submitted 05/13/2025; approved after reviewing 06/01/2025; accepted for publication 08/26/2025.