

*Е.В. Жаріков, С.Ф. Теленик*

## СПЕЦІАЛІЗОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ДИНАМІЧНОЇ КОНСОЛІДАЦІЇ ВІРТУАЛЬНИХ МАШИН

Для багатьох провайдерів хмарних послуг віртуальні машини залишаються базовою технологією віртуалізації обчислень. Віртуальні машини використовуються як для розміщення прикладних програмних засобів, так і для реалізації контейнерної віртуалізації. В умовах широкого використання віртуальних машин виникає необхідність розроблення спеціалізованого програмного забезпечення, що дозволяє визначати вплив параметрів моделей і методів управління на показники якості процесу консолідації, що дозволить запобігти виконанню експериментальних досліджень у виробничих умовах з метою оцінки нових стратегій управління ресурсами хмарного центру оброблення даних (ЦОД). Останніми роками в літературі запропоновано різні набори програмних інструментів і фреймворків для моделювання роботи ЦОД, забезпечуючи платформу та необхідні будівельні блоки для оптимізації процесу консолідації віртуальних машин. Моделі та програмні засоби моделювання процесів управління ресурсами ЦОД зазвичай не є вичерпними і вирішують конкретну проблему або завдання управління. Запропоноване у статті спеціалізоване програмне забезпечення моделювання дозволяє дослідити різні режими управління динамічною консолідацією віртуальних машин, забезпечує протоколювання результуючої інформації, такої, як показники продуктивності та діаграми навантажень, а також дозволяє визначати оптимальні параметри моделі для різних режимів роботи ЦОД, мінімізуючи кількість активних фізичних серверів та зменшуючи кількість порушень SLA.

Ключові слова: консолідація віртуальних машин, віртуалізація, хмарні обчислення, діаграма класів, діаграма послідовності.

### Вступ

Інструменти моделювання хмарних центрів оброблення даних (ЦОД) широко застосовуються для досліджень і практичного використання постачальниками хмарних послуг, щоб відтворити поведінку певної групи фізичних серверів (ФС) та віртуальних машин (ВМ) у ЦОД із застосуванням нових моделей, політик, та з метою дослідження нових алгоритмів і методів управління.

Такі програмні засоби моделювання протягом багатьох років широко використовувалися в різних дослідженнях у сфері управління ресурсами хмарних ЦОД, оскільки вивчення нових підходів із використанням виробничого середовища є ризикованим і дорогим процесом, який може істотно заважати нормальній роботі хмарних сервісів у виробничих умовах. Програмне забезпечення, яке реалізує подання системи у вигляді симуляційної моделі і поводить себе аналогічно або працює з деякими припущеннями, [1] будемо називати інструментом моделювання.

Запропонований у статті інструмент моделювання ЦОД використовує модель системи, політики управління, вхідне робоче навантаження з реального ЦОД та кількість ресурсів ФС в ЦОД. Використовуючи цю інформацію, інструмент моделювання ЦОД дозволяє моделювати процеси створення, видалення та міграції ВМ, а також отримувати такі показники якості: кількість порушень угоди про рівень обслуговування (SLA, Service Level Agreement), кількість активних ФС та кількість міграцій ВМ у процесі моделювання під час експерименту.

Іншим важливим фактором, який слід мати на увазі при моделюванні хмарного ЦОД, є кількість вхідних параметрів, які враховуються під час оцінки роботи методів управління ресурсами. Багато фреймворків і програмних засобів моделювання враховують лише робоче навантаження центрального процесора (ЦП) як вхідні дані для змодельованої ВМ. Розроблений інструмент моделювання ЦОД використовує як вхідні

дані для кожної змодельованої ВМ навантаження на чотири ресурси: ЦП, оперативну пам'ять, мережу та сховище.

Для розв'язання задачі моделювання роботи хмарного ЦОД з метою визначення близьких до оптимальних параметрів управління у статті пропонується спеціалізоване програмне забезпечення (ПЗ) моделювання ЦОД, що дозволяє досліджувати залежність показників якості управління ресурсами ЦОД відповідно до налаштувань змінних методу управління та параметрів моделі.

### 1. Аналіз публікацій

Згідно з оглядом літератури, представленим у [2, 3], інструменти та фреймворки моделювання роботи хмарного ЦОД характеризуються різними показниками, точністю моделі та архітектурною гнучкістю. Аналіз різних фреймворків моделювання показує, що надані функції і можливості недостатньо описані для розробників, які вносять модифікації з метою покращення моделі та методів управління. Крім того, при порівнянні інструментів і фреймворків моделювання слід враховувати кілька показників: функціональність, вичерпність моделі, масштабованість, вхідні дані, гнучкість, точність моделі.

CloudSim [4] – це широко використовуваний модульний і розширюваний симулятор із відкритим кодом, призначений для моделювання будь-якої можливої функціональності хмарного ЦОД та реалізації алгоритмів і методів управління для різних хмарних середовищ. Багато різних фреймворків моделювання хмарних ЦОД, запропонованих у літературі, засновані на CloudSim.

Автори [5] запропонували розширення CloudSim, щоб розробники могли внести модифікації та випробувати певну стратегію управління ресурсами з різними характеристиками та показниками якості рівня обслуговування, використовуючи георозподілені хмарні середовища. Розподілена система моделювання CloudSimScale, яка запропонована у [6], розроблена на базі CloudSim та забезпечує масштабованість системи у розподіленому середовищі, взаємодію між мо-

дулями та дозволяє моделювати процеси управління ресурсами хмарних ЦОД з використанням алгоритмів користувача. GPUCloudSim [7] – це ще одне розширення CloudSim, розроблене для аналізу та дослідження роботи політик управління з надання віртуальних машин із підтримкою GPU. Цей симулятор дозволяє моделювати розподіл ресурсів на рівні графічного процесора, щоб дослідити взаємодію між запущеними програмами користувача, накладні витрати на віртуалізацію та енергоспоживання графічних процесорів.

У [8] автори застосували моделювання дискретних подій для реалізації моделі хмарної системи, щоб дослідити продуктивність хмарної системи та отримати комбінований результат планування ВМ та відмов компонентів. Запропонований симулятор дискретних подій використовується для планування виконання програмних модулів забезпечення наукових процесів у хмарних системах та використовує продуктивність системи як цільову функцію.

Система моделювання інфраструктури як сервісу (IaaS) DISSECT-CF була запропонована у [9] і реалізує такі функції: підтримка розширюваності, підтримка оцінки споживання енергії при наданні IaaS і можливість оцінки багатьох параметрів планування та політик управління, пов'язаних із IaaS.

У [10] запропоновано систему моделювання з відкритим вихідним кодом CloudSim Plus. Автори обґрунтовують такі її переваги, як розширюваність та можливість повторного використання компонентів пропонованого фреймворку. Система моделювання забезпечує розширюваний, модульний та точний інструмент для оцінки алгоритмів управління ресурсами хмарних ЦОД.

### 2. Постановка задачі

Дослідження нових моделей і методів управління ресурсами хмарних ЦОД можна здійснити у спосіб розроблення спеціалізованих програмних засобів моделювання, які підтримують моделювання віртуалізованих систем, візуалізацію, реєстрацію показників якості управління та оцінку запропонованих алгоритмів управління на

змодельованій великомасштабній хмарній інфраструктурі.

Отже, необхідно розробити інструмент моделювання ЦОД, що забезпечує різні режими моделювання на основі гібридного підходу консолідації віртуальних машин, який об'єднує метод управління розміщенням нових віртуальних машин та метод управління міграцією активних віртуальних машин, мінімізуючи кількість активних фізичних серверів та зменшуючи кількість порушень SLA.

### 3. Розроблення моделі ЦОД з урахуванням динаміки створення, вимкнення та міграції віртуальних машин

У статті використовується модель системи, представлена в попередній статті авторів [11]. ЦОД складається з  $M$  ФС та  $N$  ВМ,  $N, M \in \mathbb{N}$ . Кожен ФС оснащений постійною кількістю ресурсів  $k$  таких, як ЦП, оперативна пам'ять, мережа та сховище,  $k \in \{CPU, RAM, NET, IO\}$  [11].

Змінні моделі визначаються так:  $c_j^k$  – необхідна ємність ресурсу  $k$  для  $j$ -ї ВМ,  $C_i^k$  – ємність ресурсу  $k$   $i$ -го ФС визначена складом обладнання,  $u_i^k$  – навантаження на ресурс  $k$   $i$ -го ФС,  $v_i$  – кількість ВМ, розміщених на  $i$ -му ФС,  $w^k \in [0, 1]$  – відносна вага ресурсу  $k$ ,  $r_{\max}^k$  – найбільша необхідна ємність кожного ресурсу  $k$  серед усіх ВМ для нормалізації,  $r_j = \sum_k (w^k c_j^k / r_{\max}^k)$  – необхідна загальна ємність ресурсів для  $j$ -ї ВМ,  $R_{\max}^k$  – найбільша наявна ємність кожного ресурсу  $k$  серед усіх ФС для нормалізації,  $R_i = \sum_k (w^k C_i^k / R_{\max}^k)$  – наявна ємність ресурсів  $i$ -го ФС,  $u_i = \sum_k (w^k u_i^k / R_{\max}^k)$  – завантаженість ресурсів  $i$ -го ФС,  $T^k \in [0, 1]$  – порогове значення доступного ресурсу  $k$   $i$ -го ФС,  $D^k \in [0, 1]$  – бажане навантаження на ресурс  $k$ ,  $L^k \in [0, 1]$  – бажане навантаження на ресурс  $k$   $i$ -го ФС, який визначений як недовантажений,  $Q^k \in (T^k, 1]$  – доступне порогове значення ресурсу  $k$   $i$ -го ФС, який може приймати мігруючі ВМ,  $v_i$  – кількість ВМ, що працюють на  $i$ -му ФС,  $d_i = \sum_k (w^k d_i^k / R_{\max}^k)$  – відхилення від бажаного навантаження  $i$ -го ФС,  $d_i^k = |c_j^k + u_i^k - C_i^k D^k|$  – відхилення від бажаного навантаження для ресурсу  $k$ ,  $f^k = (1 - T_k) \sum_{i \in B} (C_i^k - u_i^k) - c_m^k \sum_{i \in A} (v_i)$  – загальний обсяг ресурсу, який доступний для

виконання міграцій ВМ,  $c_m^k$  – ємність ресурсу  $k$ , який необхідний для забезпечення міграції ВМ,  $\theta = \sum_k (w^k \theta^k / \theta_{\max}^k)$  – оцінка можливості виконання міграції ВМ, де

$$\theta^k = \begin{cases} u_i^k < L^k C_i^k : L^k C_i^k - u_i^k \rightarrow \max \\ u_i^k \geq L^k C_i^k : -|D^k C_i^k - u_i^k| \rightarrow \max \end{cases}$$

$V$  – список нових ВМ для розміщення в системі,  $W$  – список працюючих ФС,  $W'$  – список ФС зі списку  $W$ , які мають вільні ресурси,  $R$  – список ФС, які знаходяться в режимі енергозбереження (очікування, сну)  $|W| + |R| = M$ ,  $W \cap R = \emptyset$ .

У статті авторів [11] запропоновано два методи, що забезпечують розміщення нових ВМ та міграцію працюючих ВМ. Перший запропонований метод забезпечує ефективне розміщення нових ВМ, а другий метод консолідує існуючі ВМ, застосовуючи міграції. Рис. 1 ілюструє алгоритм для першого методу початкового розміщення ВМ. Міграція ВМ використовується або для перемикавання ФС у стан енергозбереження (або стан очікування), або для розвантаження перевантажених ФС.

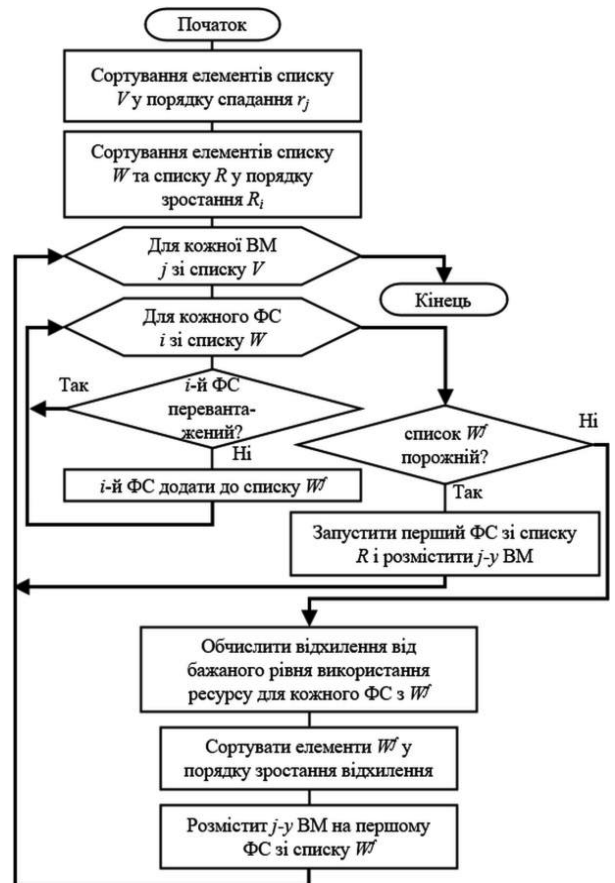


Рис. 1. Схема алгоритму розміщення нових ВМ

Міграція VM відбувається через перевантаження ФС (вільний обсяг ресурсу  $k$  менше ніж  $T^k$ ) для одного або кількох ресурсів. Перевантаження ФС призводить до нестачі ресурсів для забезпечення роботи віртуальних машин, що призводить до порушення SLA. Для кожного ФС з набору  $A$  запропонований метод шукає такий ФС з набору  $B$ , який може приймати мігруючі віртуальні машини. Тоді один або кілька ФС з набору  $A$  можна перевести в режим енергозбереження. У результаті *першого етапу методу* отримують список  $A$  та список  $B$  [11] для подальшого визначення плану міграції на *другому етапі*. Рис. 2 ілюструє алгоритм для другого методу, спрямованого на виконання консолідації віртуальних машин.

#### 4. Архітектура програмного забезпечення моделювання

Розроблений інструмент моделювання реалізований у вигляді ПЗ, яке забезпечує різні варіанти моделювання та дозволяє визначити близькі до оптимальних параметри моделі для різних режимів роботи ЦОД. ПЗ використовує багатошарову архітектуру та підхід до проектування, орієнтованого на домен (domain-driven design). Архітектура складається з: (i) рівня бази даних, що забезпечує інтерфейси для отримання даних з бази даних, (ii) рівня основної логіки та (iii) рівня подання, що забезпечує інтерфейси для виводу результатів моделювання у консоль, записування результатів у текстові файли та створення файлів MS Excel з результатами моделювання.

Основним шаблоном розроблення застосунку є шаблон проектування фасаду [12]. Він приховує складність системи та надає клієнтові інтерфейс, за допомогою якого той може отримати доступ до системи [12]. Клас фасаду відповідає за отримання даних із рівня бази даних, керує чотирма модулями та передає результати на рівень подання.

Шаблон репозиторію [13] використовується для взаємодії з рівнем бази даних, застосовуючи модель домену, щоб спростити складну бізнес-логіку та відокремити бізнес-логіку від даних. Його також можна

використовувати для доступу до джерела даних з багатьох місць і для застосування централізовано керованих правил і логіки доступу [13]. Репозиторій є посередником між рівнем джерела даних і бізнес-рівнями програмного забезпечення. Репозиторій відокремлює бізнес-логіку від взаємодії з основним джерелом даних [13].

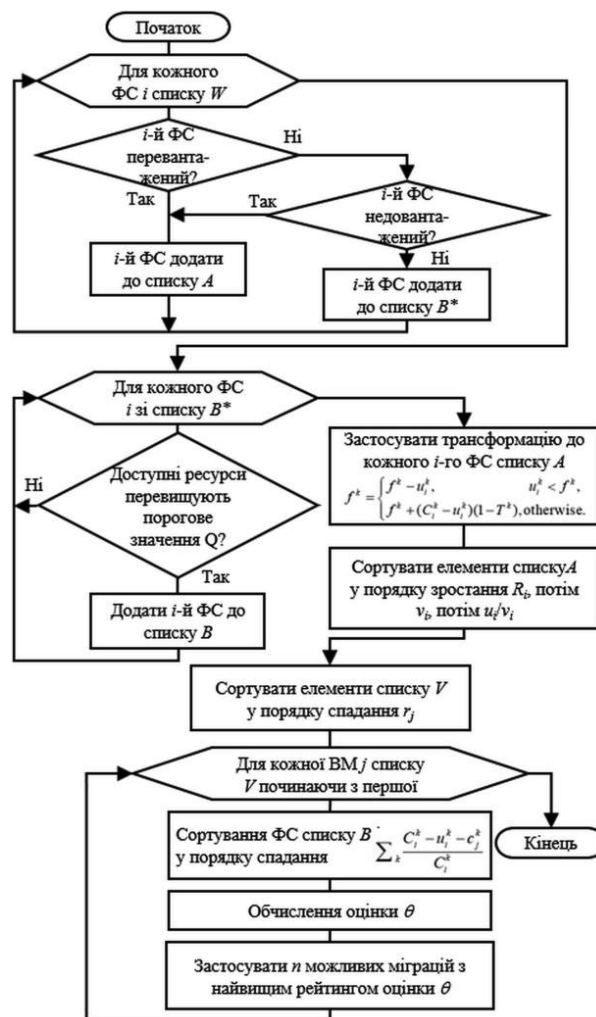


Рис. 2. Схема алгоритму консолідації VM

#### 4.1. Діаграма класів

На рис. 3 показана діаграма основних класів розробленого ПЗ моделювання. Клас Server складається з основних методів, які обробляють віртуальні машини, що знаходяться у колекції VMCollection та управляють їхнім станом. Класи колекції містять VM та ФС, які обслуговують запити під час запуску симуляції.

Клас Simulation, показаний на рис. 4, керує чотирма модулями, а саме діагностичним модулем, який виявляє перевантажені

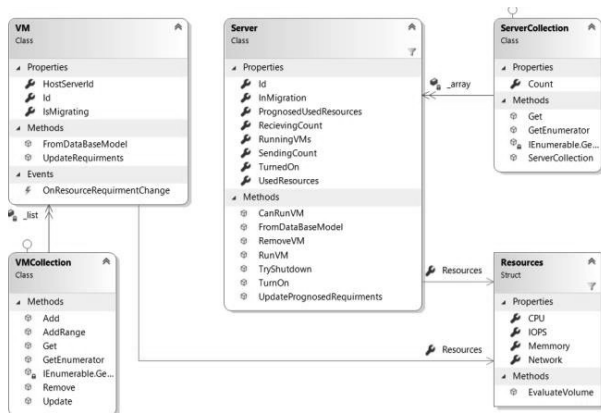


Рис. 3. Діаграма класів ПЗ моделювання

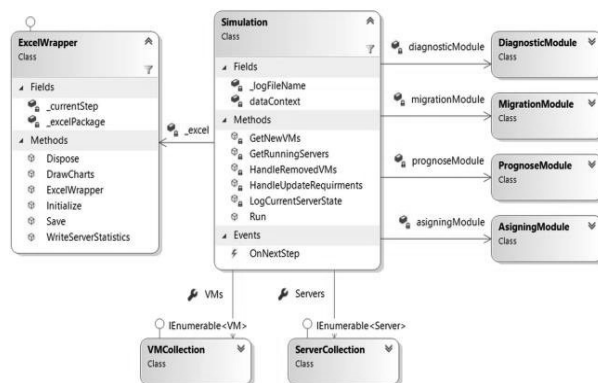


Рис. 4. Представлення класу фасаду

та недовантажені ФС, модулем розміщення нових віртуальних машин, модулем міграції, який консолідує існуючі віртуальні машини, застосовуючи міграції, і модулем прогнозування.

4.2. Діаграма послідовності

Діаграма послідовності на рис. 5 ілюструє взаємодію між основними модулями ПЗ моделювання. На першому кроці

обробляються системні події, а саме: завершення роботи ВМ, розміщення нової ВМ, оновлення вимог до ресурсів для існуючих ВМ. Потім обчислюються прогнози використання ресурсів ФС для наступних кроків моделювання (один або два кроки) за допомогою модуля прогнозування. На наступному етапі працює діагностичний модуль, визначаючи переван-

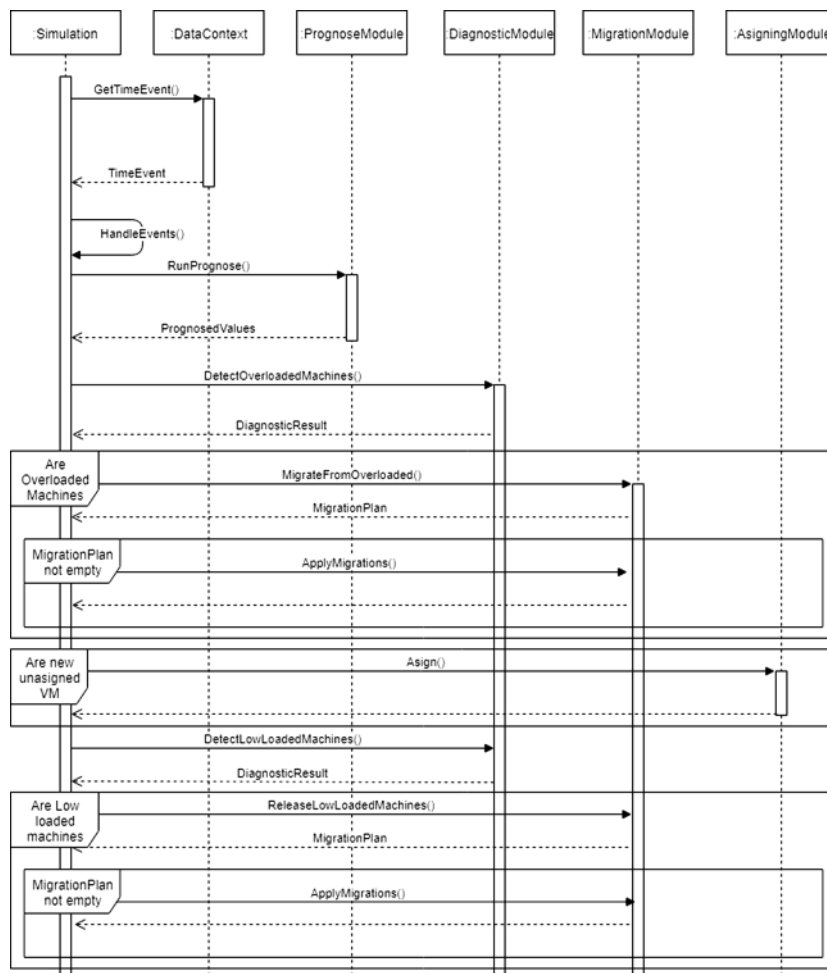


Рис. 5. Діаграма послідовності ПЗ моделювання

тажені та недовантажені ФС, які будуть передані модулю міграції ВМ для подальшого розвантаження ФС.

Якщо потрібно розмістити нові віртуальні машини, цільовий модуль виділяє багатовимірні ресурси ФС. Модуль міграції розвантажує недовантажені та перевантажені ФС. Після цього моделювання переходить до наступного кроку.

4.3. Пакети у складі ПЗ моделювання

ПЗ моделювання динамічної консолідації віртуальних машин складається з трьох пакетів, як показано на рис. 6: DAL, Utilities та Simulation. Пакет Utilities містить допоміжні класи, які використовуються в пакетах DAL і Simulation.

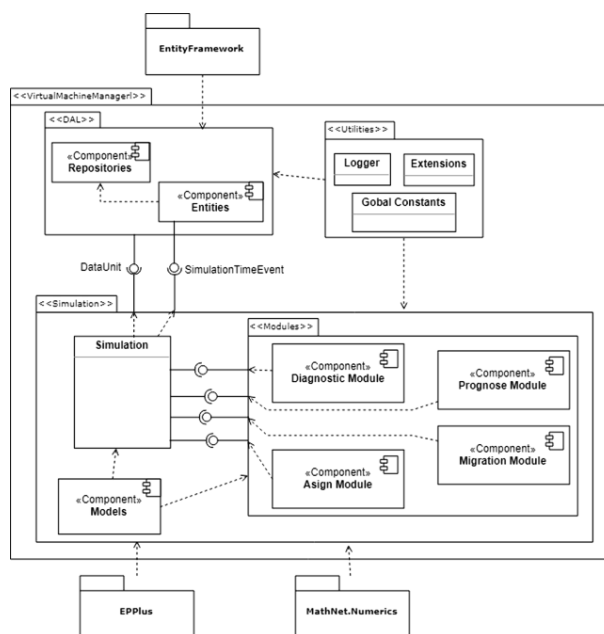


Рис. 6. Діаграма компонентів ПЗ моделювання

Пакет DAL містить логіку взаємодії з базою даних і реалізує пакет Simulation з

інтерфейсами DataUnit (містить набір репозиторіїв із відповідного компонента) і SimulationTimeEvent (основна сутність, що описує всі події на певному кроці моделювання).

Пакет Simulation містить компонент із моделями ВМ і ФС, якими керує система, і пакет із чотирма модулями, що надають інтерфейси для моделювання головного класу фасаду. Крім того, пакет DAL використовує додатковий компонент EntityFramework, а пакет Simulation використовує EPPlus і MathNet.Numerics відповідно.

5. Моделювання динамічної консолідації віртуальних машин

ПЗ моделювання для динамічної консолідації ВМ реалізовано за допомогою C#, .NET Framework 4.6 [14], SQL Server Express, ORM Framework, Entity Framework, Math.NET Numerics та EPPlus для створення звітів у файлах MS Excel за допомогою .NET [15].

Для моделювання та дослідження різних режимів роботи ЦОД на основі гібридного підходу консолідації віртуальних машин автори використовують вхідні дані Vitbrains [16]. Дані файлів Vitbrains містять записи у вигляді часових рядів про навантаження на чотири типи ресурсів  $k$ . Робоче навантаження, записане у файли Vitbrains, являє собою використання ресурсів веб-серверів, серверів баз даних або серверів додатків 1250 віртуальними машинами, які належать до 44 різних класів [17].

У процесі моделювання автори використовують 20 гетерогенних ФС із параметрами, наведеними в таблиці 1 [18]. Крім

Табл. 1. Конфігурації ФС

PM	Number	CPUMHz	The number of PEs	RAM, GB	Storage performance, IOPS	Net, Gb/s
PowerEdge R940	5	2500	112	384	32000	40
PowerEdge R740	5	2500	56	192	23190	40
PowerEdge R830	5	2200	88	256	8000	40
PowerEdge R630	5	2200	44	128	6000	40

Step	CPU			Memory			Network			IOPS			VM Count	Migrations	
	+0	+1	+2	+0	+1	+2	+0	+1	+2	+0	+1	+2		Send	Recieve
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	700	0	0	0	0	0	7000000	0	0	0	0	0	0	0	7
3	631,20	0	0	336239,72	0	0	6000005,50	0	0	3,73	0	0	2	0	6
4	7768,40	0	0	6813637	0	0	2000637,13	0	0	2782,33	0	0	6	0	2
5	8983,46	0	0	7905547	0	0	1000710,31	0	0	4018,07	0	0	7	0	1
6	10420,07	12086,28	14018,93	8459895,00	8459895,00	8459895,00	1000647,88	1000647,88	1000647,88	3678,27	2241,86	425,98	7	0	1
7	16732,26	22354,99	30793,61	10701749,00	10701749,00	10701749,00	869,21	869,21	249,00	3362,73	2792,97	2170,56	8	0	0
8	9813,06	13990,42	12886,04	33325128,00	33325128,00	33325128,00	8000598,00	229,65	117,53	2384,67	1461,28	719,50	8	8	0
9	9012,40	8688,31	8428,44	12090769,00	12090769,00	12090769,00	4000425,50	4000425,50	4000425,50	3224,27	3306,69	3166,44	4	4	0
10	259,80	789,19	625,44	2172644,50	2147479,00	2147479,00	1000004,50	1000004,31	1000004,31	75,53	-516,50	-525,45	2	1	0
11	281,80	223,20	189,34	581607,13	581607,13	581607,13	1000004,63	1000004,63	1000004,63	70,47	55,01	45,36	1	1	0
12	967,60	623,80	525,54	760563,88	760563,88	760563,88	1000008,88	1000008,88	1000008,88	56,93	42,74	33,36	1	1	0
13	402,40	326,82	192,42	1374152,88	1342173,50	1342173,50	1000004,50	1000004,50	1000004,50	31,53	25,68	16,41	4	1	0

Рис. 7. Навантаження на ресурси одного ФС

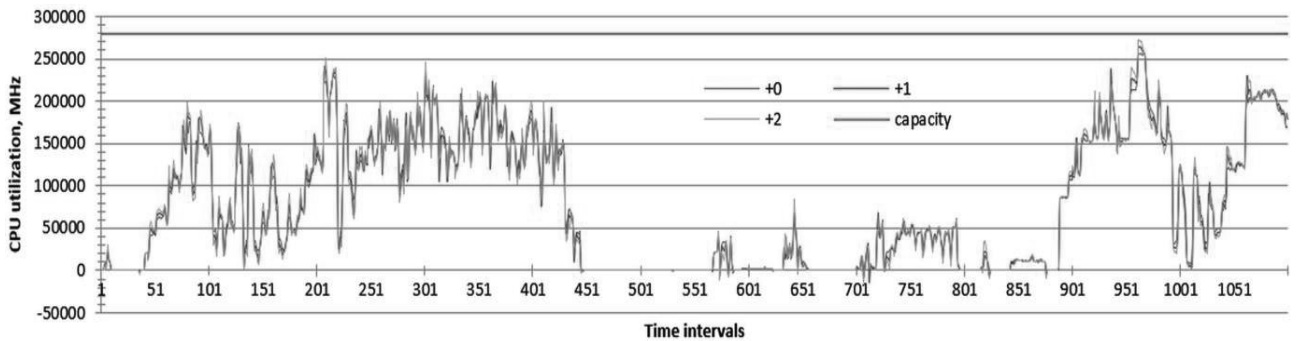


Рис. 8. Використання ЦП ФС 1 (Power-Edge R940) у файлі MS Excel

того, автори використовують дані Vitbrains, зібрані з 1114 віртуальних машин із різним обсягом запитуваного ЦП (у МГц), пам'яті (у МБ), сховища (в IOPS) та мережі (у Гбіт/с), які подаються у застосунок моделювання протягом 1100 часових інтервалів.

Параметри моделі ЦОД для кожної симуляції, показані в таблиці 2, зберігаються у файлі settings.ini.

Табл. 2. Параметри моделі

Parameter	Variable	Value
TIME_STEP_VALUE	-	300, s
STEPS_TO_SIMULATE	-	1100
RES_WEIGHT	$w^k$	1
RES_THRESHOLD	$T^k$	Changes
RES_DESIRED_LEVEL	$D^k$	Changes
RES_LOW_LEVEL	$L^k$	Changes
RES_RECIEVER_THRESHOLD	$Q^k$	Changes
BEAM_LENTH	$N$	Changes
CPU_ON_MIGRATION		100, MHz
MIN_NETWORK_ON_MIGRATION		100, Mbps

Розроблене ПЗ моделювання генерує файл MS Excel, в якому зберігається використання ресурсів, кількість розміщених VM і кількість міграцій VM для кожного ФС під час виконання симуляції, як показано на рис. 7. Файл MS Excel також містить діаграми використання усіх ресурсів під час симуляції. На рис. 8 показані фактичні, прогнозовані значення та загальна продуктивність ЦП фізичного сервера Power-Edge R940.

У застосунку використовується регресійна модель для прогнозування використання ресурсів:  $x_{t+1}^k = a_1 x_t^k + a_2 x_{t-1}^k + \dots + a_n x_{t-n+1}^k$ , де  $t$  – інтервал часу,  $n$  – кількість незалежних змінних у моделі,  $x^k$  – використання ресурсу  $k$ ,  $a$  – коефіцієнти моделі. На рис. 7 стовпець «+0» вказує на використання ресурсу на момент часу  $t$ , стовпець «+1» вказує на прогноз використання ресурсу на момент часу  $t+1$ , а стовпець «+2» вказує на прогноз використання ресурсу на момент часу  $t+2$ .

Під час першого запуску вхідні дані з папки даних Vitbrains завантажуються в базу даних, як показано на рис. 9. Дані бази даних будуть використовуватися в усіх подальших симуляціях.

```

Mapping Traces Data to DataBase Model
TimeEvents created
Reading traces
  VM 1 - done
  VM 10 - done
  VM 100 - done
  VM 1000 - done
  VM 1001 - done
  VM 1002 - done
  VM 1003 - done
  VM 1004 - done
  * * *
  VM 997 - done
  VM 998 - done
  VM 999 - done
Reading treaces - done |692ms|
Reading PM capacities...6ms
Mapping Traces Data to DataBase Model - done |410ms|
Simulation runnig
Step 1
  Updating resources requirments
    0 VMs is finished
    New VMs: 1114

```

Рис. 9. Завантаження вхідних даних у базу даних для подальшого моделювання

Під час симуляції застосунок відображає події, що відбувалися на кожному кроці. ПЗ також зберігає ці події у файлі журналу. На першому кроці ФС запускаються, як показано на рис. 10.

```

Simulation runnig
Step 1
  Updating resources requirments
    0 VMs is finished
    New VMs: 1083
  Updating resources requirments - done |982ms|
  Prognosing resources usage
  Prognosing resources usage - done |0ms|
  Diagnosting for overloaded servers
  Diagnostic - done |0ms|
  Assigning VMs
    Server 16 is turning on
    Server 17 is turning on
    Server 18 is turning on
    Server 19 is turning on
    Server 20 is turning on
    Server 6 is turning on
    Server 7 is turning on
    Server 8 is turning on
    Server 9 is turning on
    Server 10 is turning on
    Server 11 is turning on
    Server 12 is turning on
    Server 13 is turning on
  Assigning VMs - done |11ms|
Step 1 - finished |7ms|

```

Рис. 10. Перший крок симуляції

На рис. 11 показані результати оптимізації, а саме консолідація VM, за якої вдалося перемкнути три ФС (9, 10, 11) у сплячий стан. Три VM закінчили роботу і зупинені та одна VM повинна бути розміщена на якомусь ФС.

```

Step 10
  Updating resources requirments
    3 VMs is finished
    New VMs: 1
  Updating resources requirments - done |174ms|
  Prognosing resources usage
  Prognosing resources usage - done |2ms|
  Diagnosting for overloaded servers
  Diagnostic - done |0ms|
  Assigning VMs
  Assigning VMs - done |1ms|
Step 10 - finished |182ms|
Server 9 is shutting down
Server 10 is shutting down
Server 11 is shutting down
Step 11
  Updating resources requirments
    7 VMs is finished
    New VMs: 3
  Updating resources requirments - done |175ms|

```

Рис. 11. Результати оптимізації на кроці 10

План міграцій VM також відображається у консолі. На рис. 12 показано, що у процесі діагностики виявлено перевантажені ФС, через що був згенерований і застосований план міграцій для розвантаження ФС за допомогою модуля міграції. Отже, ФС 16 буде розвантажено у спосіб міграції віртуальних машин 280, 250 і 629 до ФС 3.

```

Step 16 - finished |267ms|
Server 3 is turning on
Step 17
  Updating resources requirments
    6 VMs is finished
    New VMs: 2
  Updating resources requirments - done |150ms|
  Prognosing resources usage
  Prognosing resources usage - done |1ms|
  Diagnosting for overloaded servers
  overloaded servers detected at prognose level 0
  Diagnostic - done |1ms|
  Migration Plan contains 3 migrations
  VM   From  To
  608   8      3
  610   8      3
  711   8      3
  Assigning VMs
  Assigning VMs - done |0ms|
Step 17 - finished |171ms|

```

Рис. 12. Згенерований план міграцій на кроці 17

## 6. Оцінка результатів моделювання

Під час запусків застосунку моделювання встановлено, що наявна сумарна потужність двадцяти ФС, обраних для симуляції роботи ЦОД з метою розміщення 1114 VM, є недостатньою, про що свідчить значна кількість міграцій VM.

У багатьох випадках немає необхідності визначати адаптивні порогові значення для використання певних ресурсів, оскільки експерименти показують, що агресивна по-



літика перемикання ФС у режим енергозбереження не допомагає мінімізувати кількість порушень SLA. Натомість адаптивні порогові значення рекомендується визначати відповідно до наявності стрибків робочого навантаження. Чим частіше такі стрибки, тим менше порогове значення треба використовувати. Крім того, чутливість багатьох параметрів моделі до невеликого діапазону змін істотно не впливає на показники якості.

Запропоновані методи динамічної консолідації ВМ дозволяють ефективно розвантажувати ФС, щоб перевести їх у сплячий стан, зменшуючи споживання електроенергії. Результати порівняння, представлені в [11], показують, що запропоновані методи перевершують Best Fit алгоритм щодо порушення SLA (зменшення на 46,7%), кількості активних ФС (зменшення на 33,2%) та кількості міграцій ВМ (зменшення на 42,8%).

### Висновки

Останні дослідження консолідації віртуальних машин виявили необхідність дослідження нових політик управління за допомогою інструментів моделювання хмарного ЦОД з використанням реалістичного робочого навантаження.

Розроблено спеціалізоване програмне забезпечення моделювання ЦОД для реалізації підходу до динамічної консолідації віртуальних машин, що надає можливість проведення симуляцій із використанням вхідних даних, отриманих з реального ЦОД. Запропоноване програмне забезпечення дозволяє досліджувати різні режими управління ресурсами ЦОД на основі гібридного підходу консолідації віртуальних машин, який об'єднує метод управління розміщенням нових віртуальних машин та метод управління міграцією активних віртуальних машин, мінімізуючи кількість активних фізичних серверів та зменшуючи кількість порушень SLA.

У статті представлена архітектура програмного забезпечення, діаграми класів, діаграма послідовності, пакети програмного забезпечення та додаткові інструменти реалізації функцій. Експерименти показують, що запропонований програмний застосунок забезпечує широкий

спектр реєстраційної та налагоджувальної інформації за допомогою текстових файлів та файлів MS Excel із діаграмами, забезпечує різноманітні види моделювання та дозволяє визначити оптимальні параметри моделі для різних режимів роботи з точки зору порушення SLA, кількості активних ФС та кількості міграцій ВМ.

### References

1. IEEE Std 610.3-1989. *IEEE Standard Glossary of Modeling and Simulation Terminology*. Institute of Electrical and Electronic Engineers (IEEE), New York, NY, 1989.
2. A. Ismail, «Energy-driven cloud simulation: existing surveys, simulation supports, impacts and challenges,» *Cluster Computing*, vol. 23, pp. 3039–3055, 2020.
3. N. Mansouri, R. Ghafari, and B. Mohammad Hasani Zade, «Cloud computing simulators: A comprehensive review,» *Simulation Modelling Practice and Theory*, vol. 104, pp. 102-144, 2020.
4. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya, «CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,» *Software: Practice and experience*, vol. 41, no. 1, pp. 23-50, 2011.
5. H. Jeon, C. Cho, S. Shin, S. Yoon, «A CloudSim-Extension for Simulating Distributed Functions-as-a-Service,» *20th International Conference on parallel and distributed computing, applications and technologies (PDCAT)*, 2019, pp. 386–391.
6. B. Elahi, A. W. Malik, A. U. Rahman, M. A. Khan, «Toward scalable cloud data center simulation using high-level architecture,» *Software: Practice and Experience*, vol. 50, no. 6, pp. 827–843, 2020.
7. A. Siavashi, M. Momtazpour, «GPUCloudSim: an extension of CloudSim for modeling and simulation of GPUs in cloud data centers,» *The Journal of Supercomputing*, vol. 75, no. 5, pp. 2535-2561, 2019.
8. D. Oliveira, A. Brinkmann, N. Rosa, «Performance Evaluation and Optimization of Workflow Applications in Cloud Environments,» *Journal of Grid Computing*, vol. 17, no. 4, pp. 749–770, 2019.
9. G. Kecskemeti, «DISSECT-CF: a simulator to foster energy-aware scheduling in infrastructure clouds,» *Simulation Modelling Practice and Theory*, vol. 58, pp. 188-218, 2015.

10. M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. Inácio, and M. M. Freire, "CloudSim plus: a cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, IEEE, 2017, pp. 400-406.
11. E. Zharikov, S. Telenyk, O. Rolik, and Y. Serdiuk, "Cloud resource management with a hybrid virtual machine consolidation approach," *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, 2019, pp. 289-294.
12. Design Patterns - Facade Pattern, [online] Available: [https://www.tutorialspoint.com/design\\_pattern/facade\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/facade_pattern.htm)
13. The Repository Pattern <https://msdn.microsoft.com/en-us/library/ff649690.aspx>
14. .NET Framework documentation, [online] Available: <https://docs.microsoft.com/en-us/dotnet/framework/>
15. Interoperability Overview, [online] Available: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/interop/interoperability-overview>
16. GWA-T-12 Bitbrains, [online] Available: <http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains>
17. S. Shen, V. V. Beek and A. Iosup, "Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters," *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, 2015, pp. 465-474.
18. PowerEdge Rack Servers, [online] Available: <https://www.dell.com/en-us/work/shop/dell-poweredge-servers/sc/servers>

Отримано 07.03.2022

### ***Про авторів:***

Жаріков Едуард В'ячеславович,  
доктор технічних наук,  
професор кафедри інформатики  
та програмної інженерії Національного  
технічного університету України  
«Київський політехнічний інститут  
імені Ігоря Сікорського».  
Кількість наукових публікацій  
в українських виданнях - 125.  
Кількість наукових публікацій  
в зарубіжних виданнях - 37.  
Індекс Гірша - 4.  
<http://orcid.org/0000-0003-1811-9336>,

Теленик Сергій Федорович,  
доктор технічних наук, професор,  
декан факультету інформатики  
та обчислювальної техніки Національного  
технічного університету України  
«Київський політехнічний інститут  
імені Ігоря Сікорського».  
Кількість наукових публікацій  
в українських виданнях - 325.  
Кількість наукових публікацій  
в зарубіжних виданнях - 67.  
Індекс Гірша - 4.  
<https://orcid.org/0000-0001-9202-9406>,

### ***Місце роботи авторів:***

Національний технічний університет  
України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Тел.: 38044 204 86 10,  
e-mail: [zharikov.eduard@acts.kpi.ua](mailto:zharikov.eduard@acts.kpi.ua)