

УДК 004.78

Р.О. СИНКЕВИЧ\*, С.Я. МАЙСТРЕНКО\*, Т.О. ДОНЦОВ-ЗАГРЕБА\*, О.В. ХАЛЧЕНКОВ\*,  
О.О. ПОЛОНСЬКИЙ\*, О.І. УДОВЕНКО\*

## ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ АТМОСФЕРНИХ ЗАБРУДНЕНЬ У ХМАРНІЙ ІНФРАСТРУКТУРІ ШЛЯХОМ АВТОМАТИЗАЦІЇ СТВОРЕННЯ ВІРТУАЛЬНИХ МАШИН

\*Інститут проблем математичних машин і систем НАН України, м. Київ, Україна

**Анотація.** У роботі вдосконалено веб-систему прогнозування атмосферних забруднень у випадку техногенних аварій «Повітря» шляхом збереження налаштованої моделі атмосферного перенесення CALPUFF у попередньо налаштованому образі та інтеграції в системі програмних засобів автоматичного створення віртуальних машин із попередньо збереженого образу після запиту користувача на розрахунок. Для цього в Українській національній грід інфраструктурі було налаштовано образ на базі операційної системи Ubuntu разом із необхідними бібліотеками моделі атмосферного перенесення CALPUFF. На базі збереженого образу можна швидко створювати і запускати віртуальну машину і розпочинати розрахунок. Після розрахунку система «Повітря» видаляє віртуальну машину, звільнюючи ресурси хмари для інших задач. На прикладі тестових розрахунків показано, що завдяки винесенню модуля розрахунку атмосферного перенесення у хмарну інфраструктуру вдалося зменшити час розрахунку моделі CALPUFF майже у 2 рази, а час завантаження даних прогнозу погоди зменшено більше ніж у 10 разів. Сумарно час виконання розрахунку системи після запиту користувача зменшився для тестової конфігурації у 4 рази. Крім того, підвищилась стійкість системи щодо збоїв у роботі хмарної інфраструктури. Створений архів з операційною системою та налаштованим для роботи моделі CALPUFF за потреби також можна переносити на інші приватні хмари (наприклад, Amazon, Google Cloud Platform, Microsoft Azure) та запускати віртуальні машини. Таким чином, розроблена веб-система відповідає сучасним трендам із впровадження хмарних технологій, дозволяє за необхідності масштабування та може бути адаптованою для інших приватних чи публічних обчислювальних хмар. Система доступна для зареєстрованих користувачів за посиланням: [http://cloud-2.bitp.kiev.ua/airsystem\\_english/airsystem.html](http://cloud-2.bitp.kiev.ua/airsystem_english/airsystem.html).

**Ключові слова:** атмосферні забруднення, хмарні обчислення, грід інфраструктура, інформаційні системи підтримки рішень, розповсюдження атмосферних забруднень, CALPUFF.

**Abstract.** The paper improves the web service Povityra aimed at forecasting air pollution in case of man-made disasters by saving the configured CALPUFF atmospheric transport model in a pre-configured image and integrating into the system of software tools for the automatic creation of virtual machines from a pre-saved image after the user request. To do this, an image based on the Ubuntu operating system was configured in the Ukrainian National Grid Infrastructure, together with the necessary CALPUFF atmospheric transport model libraries. Based on the saved image, one can quickly create and run a virtual machine and start the calculations. After the calculation, the Povityra system removes the virtual machine, freeing up cloud resources for other tasks. The example of test calculations shows that due to the transfer of the module of calculation of atmospheric transport to the cloud infrastructure it was possible to reduce the calculation time of the CALPUFF model by almost 2 times, and the download time of weather forecast data was reduced by more than 10 times. In total, the system calculation time after the user's request was reduced by 4 times for the test configuration. In addition, the resilience of the system to cloud infrastructure failures has been increased. The created archive with the operating system and configured for the CALPUFF model runs can also be transferred to other private clouds (for example, Amazon, Google

Cloud Platform, Microsoft Azure) and their virtual machines could be used. Thus, the developed web system corresponds to modern trends in the implementation of cloud computing technologies, if necessary, allows scaling and can be adapted to other private or public cloud computing infrastructures. The system is available for registered users by the link: [http://cloud-2.bitp.kiev.ua/airsystem\\_english/airsystem.html](http://cloud-2.bitp.kiev.ua/airsystem_english/airsystem.html).

**Keywords:** air pollution, cloud computing, grid infrastructure, decision support systems, spread of air pollution, CALPUFF.

DOI: 10.34121/1028-9763-2022-2-68-76

## 1. Вступ

Концепція хмарних обчислень з'явилася відносно недавно, проте за короткий час вона стала найгучнішим трендом на ринку інформаційних технологій (ІТ). Насправді, хмарні обчислення є одним із варіантів реалізації відомої концепції розподілу часу над ресурсами з урахуванням сучасного розвитку технічних засобів. Завдяки принципу використання апаратних ресурсів за розподілом часу можна домогтися більшої ефективності використання хмарних обчислень, звільнювати ресурси для інших задач у хмарі, коли вони не використовуються, спростити налаштування системи для розрахунків і забезпечити легку міграцію до іншого хмарного провайдера.

У 2009 році Кабінетом Міністрів України було прийнято програму щодо створення і впровадження грід технологій на 2009–2013 роки, завдяки чому була реалізована платформа хмарних обчислень на основі хмарного кластера Українського національного гріду (УНГ) [1]. Загальна обчислювальна потужність хмарного кластера складає 10 обчислювальних серверів, 576 віртуальних ядер, 2560 Gb оперативної пам'яті, 200 Tb дискової пам'яті. Хмарні ресурси УНГ відповідають сучасним вимогам доступності та надійності, що дозволяє ефективно використовувати обчислювальні ресурси, реалізуючи модель «виділення ресурсів за вимогою».

Платформа хмарних обчислень УНГ призначена не тільки для виконання складних обчислювальних задач, але й насамперед для створення наукоємних сервісів загального користування, призначених для наукової та експертної спільноти України, на підставі яких формується веб-каталог хмарних сервісів Віртуального центру цифрової науки (ВЦЦН) на базі хмарного кластера УНГ ([http://cloud-5.bitp.kiev.ua/?page\\_id=21](http://cloud-5.bitp.kiev.ua/?page_id=21)). Планується майбутня інтеграція веб-сервісів УНГ з Європейською хмарою відкритої науки (<https://opensciencegrid.org/>).

Відповідно, у попередніх роботах авторів веб-сервіс прогнозування атмосферних забруднень “Повітря” [2] було налаштовано на одній із віртуальних машин (ВМ) УНГ та додано його до каталогу сервісів [3]. У такій реалізації є низка недоліків, пов'язаних із відмовостійкістю та довгим часом повторного налаштування машини у разі проблем із ВМ, на якій проводиться розрахунок.

Отже, метою даної роботи є створення та інтеграція у веб-сервісі прогнозування атмосферних забруднень програмних засобів для автоматичного запуску віртуальної машини після запиту користувача на розрахунок, що дозволить удосконалити розроблений раніше веб-сервіс.

## 2. Автоматизація створення ВМ у платформі хмарних обчислень УНГ

Створення нової віртуальної машини у хмарній інфраструктурі можна умовно розділити на декілька етапів, послідовне виконання яких дозволить створити віртуальну машину відповідно до потрібних характеристик і параметрів для підключення до неї. Основним завданням є виділення необхідних ресурсів і запуск віртуальної машини. Для виконання цього завдання можна використати програмний компонент OpenStack – Nova [4], який дозволяє розподіляти ресурси між ВМ (або інстанс від англ. instance), використовуючи готові шаб-

лони. Він дозволяє підібрати потрібні параметри віртуальних ресурсів, забезпечити безпечний доступ до машини через мережу інтернет та легко вносити зміни.

Для запуску нової VM команді *nova boot* потрібно передати такі параметри: апаратні характеристики (*flavor*), образ операційної системи (*image*), унікальний ключ доступу (*key-name*), групу безпеки (*security-groups*) – список відкритих портів у *firewall* і останній параметр – унікальна назва віртуальної машини. Частина параметрів, таких як характеристика інстансу, образ операційної системи і група безпеки вже є на сервері у вигляді шаблонів з унікальним іменем та ID, які можна використати як параметри без змін. Приклад повного коду для запуску VM:

```
nova boot --flavor 5\  
--image 97317539-ebc4-4be4-aa0f-23c9b5139639\  
--key-name calpuff_m1_large_key\  
--security-groups d134acb2-e6bc-4c82-a294-9617fdf7bf07\  
calpuff_m1_large
```

Перший параметр – це апаратні характеристики, які будуть виділені перед запуском VM. Вони зберігаються у вигляді шаблонів, які можна розширювати власними. Для того, щоб показати доступні варіанти характеристик, потрібно виконати команду: *nova flavor-list* (рис. 1). Для використання одного з шаблонів необхідно запам'ятати його ID або повне ім'я і передати його при запуску як перший параметр команді *nova boot*.

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public
0	m1.tiny	512	10	0		1	1.0	True
1	m1.small	1024	20	0		1	1.0	True
2	m1.medium	2048	40	0		2	1.0	True
3	m1.large	4096	80	0		2	1.0	True
4	m1.xlarge	8192	160	0		4	1.0	True
5	c3.2xlarge	8192	16	0		6	1.0	False
5f805b2e-8488-4560-90c4-ee4bf16cb559	m1.condor12	24576	50	0		12	1.0	False
8e3f6200-3957-4582-8271-afd7e14aa3d2	m1.condor24	49152	50	0		24	1.0	False
c7c1ef6b-bb73-4395-91be-dddb71878b21	m3.large	4096	40	0		4	1.0	True
u56ee93c-4034-4399-9a82-7064a0488876	m2.large	4096	20	0		4	1.0	True
da78fb03-facc-4eb9-a163-b994ce07bd75	romans_hello_world	4096	10	0		1	1.0	True

Рисунок 1 – Доступна конфігурація віртуального обладнання в УНГ

Другий параметр – *image* визначає архів з операційною системою, яка буде використовуватися. Для того, щоб перевірити, які образи операційних систем є на сервері (з фільтром Ubuntu), необхідно виконати команду *openstack image list | grep Ubuntu* (рис. 2).

ID	Name	Status
a4ffb73f-ecc8-40a8-8195-a024f275f432	Image for EGI Docker [Ubuntu/18.04/VirtualBox]	active
a37a06b3-fb02-4d7d-8b8e-f832acfaeb2	Image for EGI Small Ubuntu 16.04 for Monitoring [Ubuntu/16.04/VirtualBox]	active
29c43061-105c-4c0a-b9ea-fe070a14dcff	Image for ScipionCloud v3.0 [Ubuntu/18.04/VirtualBox]	active
6952b8ca-090e-4df2-b7ff-3d93c6d1439a	Image for ScipionCloud v3.0 [Ubuntu/18.04/VirtualBox] with GPU support	active
5436faef-6119-4ff3-8530-fb19b07e59ef	Image for Ubuntu Server 16.04 LTS [Ubuntu/16.04/KVM]	active
4ae5903a-b005-440c-aa80-17b0f3bde13a	Image for Ubuntu Server 18.04 LTS [Ubuntu/18.04/KVM]	active
ccb5efca-a9b0-4642-b35c-aa322ba5c3b1	Image for Ubuntu Server 20.04 LTS [Ubuntu/20.04]	active
d9657d95-4800-485b-b082-3e74efe17e9f	Ubuntu 16.04 server plus	active
97317539-ebc4-4be4-aa0f-23c9b5139639	Ubuntu 18.04 server plus	active
5c4ceaf0-2d65-46a0-b62d-4be163804809	Ubuntu 20.04 server plus	active
0d11296d-99f8-4aec-a39c-c18d1c05f659	Ubuntu-cosmic	active

Рисунок 2 – Шаблони операційної системи Ubuntu у хмарі ВІТР

Третій параметр *key-name* – це назва ключа доступу до VM. Кожен ключ має бути унікальним. При цьому його назва повинна бути зрозумілою та містити в назві асоціації з

назвою машини для того, щоб його було простіше знайти у разі необхідності ручного підключення до VM.

Наступний параметр – група доступу *security-groups* містить інформацію про доступ. Подивитися наявні групи доступу можна за командою *openstack security group list* (рис. 3). Кожна група незалежно від імені має унікальний ID, який і використовується для запуску нової VM.

```

@0:551:roman@cloud-2:~$ openstack security group list

```

ID	Name	Description	Project
06c88520-2051-45d4-b4d0-f61bca315cbf	default	Default security group	8d48558b378549a3a3f02ce033096c1f
11fa08a0-5c99-4467-b5d2-b51ed1ed700d	default	Default security group	8d07a86bfac04320bc8cce309699ef8d
15030fe0-1f1d-4a5d-9376-2a274e7b55db	default	Default security group	e3c79a75db8543a6ae4c73dba559e916
1cc3d203-d5a7-4b13-8dd6-06449b3eda1f	CALPUFF_group	for CALPUFF instance	4e533496acf04893ac21fffebb44e361
1d3f1d98-f1cb-4eac-99fb-3998dad70761	global_http	Allows Web traffic anywhere on the Internet.	4e533496acf04893ac21fffebb44e361
46765d40-5ed5-4e0b-b2fc-908d4ed1c4ff	IMMSP	for running IMMSP instances (NRF4IK, etc)	5fa6916e14f044f0be9364eca33a6176
483504ae-7a59-46a8-9c13-9b7532e09309	default	Default security group	
4d90a572-cc71-4d27-a20a-5d50684600e6	default	Default security group	60691b125717491a841ac1599ab6b3a1
52bd223a-1326-4266-841e-9288cd05c02f	white_listed	Full access from selected IP/range	04e98f781311489aafbfe6901f42e9d6
59b2e802-8e05-4978-a6e5-dbfeca781aec	default	Default security group	a37b6ef83e8441e6adfce4455c3fbb96

Рисунок 3 – Збережені шаблони групи доступу для інстансів в УНГ

Кожне створення нової машини разом із обранням системи потребує генерацію унікальної пари ключів, які будуть використані для доступу до неї. Тому першим завданням є створення унікальної пари ключів для створення VM (public key) та доступу до неї після її запуску (private key). Кожен ключ має створюватись з унікальним іменем. Для цього добре підходять дата і час на момент створення VM, що додаються до префіксу “private-key”.

Приватний ключ зберігається у прихованій підпапці “.ssh” у папці проєкту, наприклад, */home/roman/vm*.

Код створення нового ключа:

```

KEY_NAME="private-key-`date --utc +%Y%m%d%H%M`"
FILE_PATH=.ssh/"${KEY_NAME}.key"
openstack keypair create $KEY_NAME >> $FILE_PATH
chmod 600 .ssh/"${KEY_NAME}.key"

```

Наступним кроком є створення змінної з унікальним іменем інстансу. Це необхідно для підключення та моніторингу його серед інших VM. Іменем було обрано ім'я користувача з додаванням суфіксу “\_vm\_” та поточної дати і часу. Використовуючи це унікальне ім'я разом з іншими базовими параметрами, ми можемо створити декілька різних VM, запускаючи один і той же скрипт декілька разів підряд.

Код створення нового інстансу:

```

VMNAME="${USER}"_vm_`date --utc +%Y%m%d%H%M%S`
nova boot --flavor 5
--image 97317539-ebc4-4be4-aa0f-23c9b5139639\
--key-name $KEY_NAME\
--security-groups d134acb2-e6bc-4c82-a294-9617fdf7bf07\
$VMNAME 2>/dev/null

```

Останнє завдання – це очікування повного запуску VM та виведення інформації з командою доступу до інстансу. Виділення ресурсів у віртуальному пулі і установка системи вимагають певного часу, який наперед неможливо визначити. Тому було створено цикл із таймером, який перевіряє готовність системи протягом часу, заданого у змінній *TIMER*. Шляхом дослідження виявлено, що оптимальним часом для перевірки є 1 хв (60 с). Якщо при виконанні виникнуть проблеми (наприклад, таке ім'я вже використано або недостат-

ньо ресурсів) і інстанс не буде запущено протягом 3 хвилин, то його буде видалено. Повний код скрипта доступний на веб-сервісі GitHub [5].

При тестовому запуску скрипта на ресурсах УНГ для послідовного створення 100 VM в усіх випадках VM були успішно створені. При цьому у 73% випадків VM створювались із першої ітерації.

### 3. Інтегрування програми автоматизації створення VM у системі прогнозування забруднень

Для забезпечення виконання обчислень на основі моделі атмосферного перенесення забруднень CALPUFF [6] у складі веб-системи «Повітря» [2] у хмарному середовищі необхідно провести інтеграцію розробленої програми автоматизованого створення віртуальної машини після запиту користувача на розрахунок та модуля безпосереднього формування прогнозу.

#### 3.1. Архітектура системи

Для спрощення процесу формування прогнозу в даній архітектурі передбачено такі кроки (рис. 4):

- 1) віддалені користувачі формують набір параметрів прогнозу та передають до веб-серверу запит на формування прогнозу;
- 2) на веб-сервері виконується запуск скрипта створення віртуальної машини, який після виконання повертає параметри доступу до VM у `config.json` файлі;
- 3) у разі успішного виконання попередньої дії проводяться завантаження відповідного прогнозу погоди на VM та послідовний запуск моделей CALPUFF для формування прогнозу забруднення;
- 4) результати сформованого прогнозу передаються з віртуальної машини до БД системи і можуть бути переглянуті та проаналізовані користувачем;
- 5) після завершення передачі результатів прогнозу до веб-серверу виконується видалення створеної віртуальної машини та файлів, пов'язаних із нею.

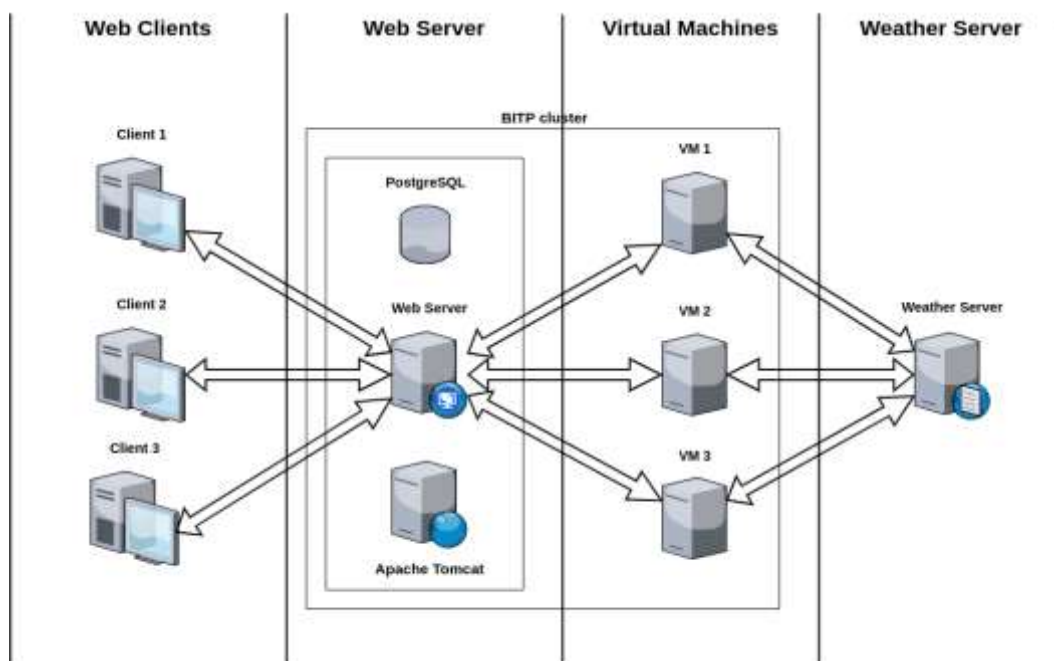


Рисунок 4 – Схема роботи CALPUFF із хмарою BITP

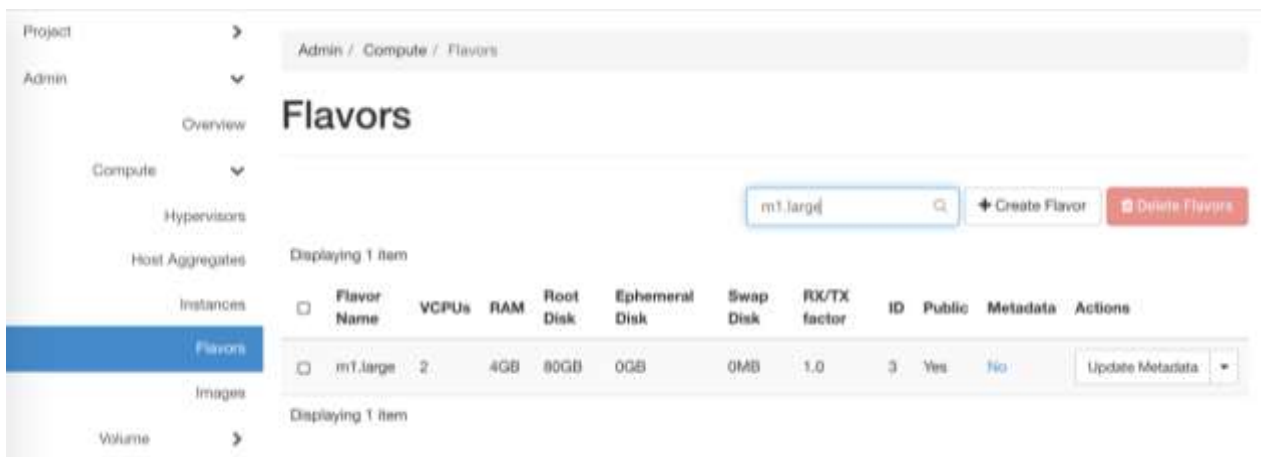
### 3.2. Шаблон фізичних ресурсів для виконання розрахунків

Розрахунки моделі атмосферного перенесення CALPUFF проводяться на створеній VM. Ресурси, які виділяються для створеної VM, мають бути визначені заздалегідь. Кожний інстанс використовує певний набір «фізичних ресурсів», які, на відміну від справжніх, можна легко змінювати, додаючи чи зменшуючи потрібну їх кількість. Слід зазначити, що розробникам доступна лише частина ресурсів УНГ. У даному випадку це 50 віртуальних ядер, 100 Gb оперативної пам'яті, 9,8 TB дискової пам'яті та до 10 одночасно працюючих VM. Але оскільки розробники використовують хмарну платформу УНГ і для інших задач, окрім розробки веб-системи прогнозування забруднень, для системи використовується лише частина наявних ресурсів.

Для створення нового шаблону необхідно команді *openstack flavor create* передати оптимальні параметри фізичних ресурсів у залежності від задачі. Приклад команди для створення шаблону віртуального обладнання:

```
openstack flavor create FLAVOR_NAME --id FLAVOR_ID --ram RAM_IN_MB --disk ROOT_DISK_IN_GB --vcpus NUMBER_OF_VCPUS
```

У даній роботі було обрано популярний шаблон під назвою «*m1-large*», що за своїми характеристиками включає два віртуальні ядра, 4Гб оперативної пам'яті та 80Гб постійної (дискової) пам'яті, що максимально підходить для даної задачі (рис. 5).



Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
m1.large	2	4GB	80GB	0GB	0MB	1.0	3	Yes	No	Update Metadata

Рисунок 5 – Характеристики шаблону «m1-large» у хмарі ВІТР

### 3.3. Образ VM для виконання розрахунків

Робота будь-якої VM базується на певній операційній системі. Зазвичай для хмарних розрахунків – це дистрибутив на базі операційної системи Linux. Щоб прискорити запуск системи, операційну систему зберігають у вигляді попередньо налаштованого образу, який зберігається окремо у хмарі.

Для того, щоб підготувати image з CALPUFF, спочатку було запущено VM на базі Ubuntu 16.04, де після підключення до неї була налаштована і протестована модель CALPUFF на тестових даних. Після цього VM виключається, і на її базі створюється image шляхом передачі команді *nova image-create* імені створеної VM або її унікального ідентифікатора та нового імені, яке буде використане для image. Характеристики образу VM із налаштованим CALPUFF під назвою «calpuff\_snapshot» показано на рис. 6.

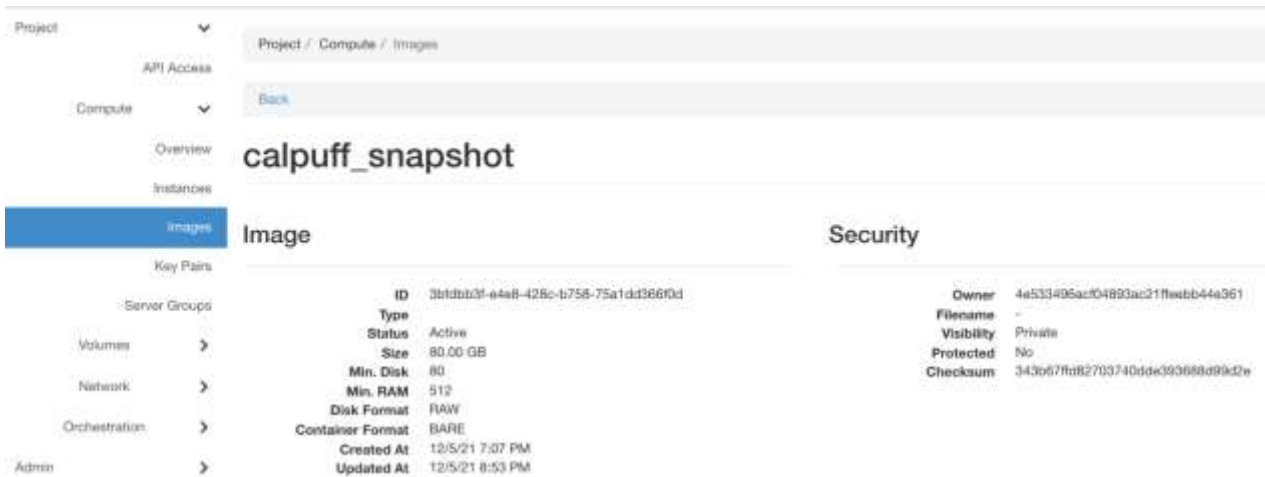


Рисунок 6 – Збережений образ VM із налаштованим CALPUFF у хмарі ВІТР

### 3.4. Скрипт створення VM у складі веб-системи

Скрипт, який перевіряє можливість створення нової машини та її запуск, представлений на рис. 7. Скрипт дозволяє створювати віртуальну машину з унікальним іменем, яке складається із трьох частин: назва задачі, для якої створена машина – CALPUFF; назва шаблону фізичних ресурсів – *ml-large*; хеш-значення – для уникнення проблем із запуском та видаленням двох однакових машин із однаковими назвами.

Основним файлом, який використовується для зв'язку між веб-системою та створеною машиною, є *config.json*, що динамічно змінюється в залежності від статусу роботи VM (створено, видалено).

```

[17:25] @openstackcloud:~/home/roles/ml - cat launch_calpuff_vm.sh
#!/bin/bash

TIME=$(date +%d.%m.%Y-%H:%M:%S)

echo "TIME starting creating VM" >> launching.log

i=/home/rozan/WRF-UNG.rc
mkdir -p $i

BASH="date --utc +%Y%m%d%H%M%S"; FLAVOR="ml_large"; TIMER=60

VM_NAME="calpuff_${FLAVOR}/${i}_$(HASH)";
KEY_PATH="ssh/${VM_NAME}.key";
openstack keypair create $VM_NAME >> $KEY_PATH
chmod 600 "$i/ssh/${VM_NAME}.key"

while true; do
  nova boot --flavor $FLAVOR \
    --image 3bfdbb3f-e4e8-428c-b758-75a1dd366f0d \
    --key-name $VM_NAME \
    --security-groups d134ac2-e6bc-4cb2-a294-9617fd76f97 \
    $VM_NAME 2>/dev/null
  for i in `seq 1 3`; do
    sec=$TIMER
    while [ $sec -ge 0 ]; do
      echo -ne "$i attempt to start VM: $sec@03310Kv"
      let "sec=sec-1"
      sleep 1
    done

    STATUS=$(openstack server list | grep $VM_NAME | awk '{ print $6 }')
    IP=$(openstack server list | grep $VM_NAME | awk '{ split($8, v, "."); print v[2] }')
    SYSTEM=$(openstack server list | grep $VM_NAME | awk '{ print $10 }')

    if [ "$STATUS" = "ACTIVE" ]; then
      printf "VM $VM_NAME is $STATUS, IP address $IP, system $SYSTEM\n"
      printf "To connect use: ssh -i $KEY_PATH ubuntu@$IP\n"
      echo "TIME VM $VM_NAME is $STATUS, IP address $IP, system $SYSTEM" >> launching.log
      echo -e "TIME To connect use: ssh -i $KEY_PATH ubuntu@$IP\n" >> launching.log
      echo -e "[\n  \"hostname\": \"$VM_NAME\", \n  \"ip\": \"$IP\", \n  \"status\": \"$STATUS\" \n]\" > config.json
      exit
    fi
  done
  echo -e "[\n  \"hostname\": \"$VM_NAME\", \n  \"ip\": \"$IP\", \n  \"status\": \"$STATUS\" \n]\" > config.json
  printf "Trying to delete VM $VM_NAME with $STATUS status, IP address $IP, system $SYSTEM\n"
  openstack server delete $(openstack server list | grep $VM_NAME | awk '{ print $2 }')
done

```

Рисунок 7 – Скрипт запуску віртуальної машини

#### 4. Тестування розрахунків у хмарній інфраструктурі

Для оцінки ефективності використання хмарних обчислень були проведені обчислювальні експерименти. Формування прогнозу виконувалось для області з координатами південно-західного кута 47,60616 по широті та 26,10352 по довготі, 40 комірок по широті та довготі із кроком 20 км. Координати джерела забруднення 50,5633 по широті та 30,4778 по довготі, забруднювач – хлор у кількості 100 т, термін дії джерела 1 год.

Виміри часу виконання виконувались окремо для завантаження необхідного прогнозу погоди з серверу погоди та безпосередньо для виконання обчислень. Необхідно зазначити, що час завантаження файлу із прогнозом погоди залежить від швидкості інтернету в мережі і може сильно коливатись у різні моменти часу. Але проведення обчислень залежить від технічних характеристик веб-серверу та VM і не буде залежати від моменту формування прогнозу.

У проведеному експерименті при формуванні прогнозу на веб-сервері (рис. 8) час завантаження прогнозу погоди становив 107 хв, а при проведенні розрахунків у хмарному середовищі – всього 7 хв. Час формування прогнозу в першому випадку становив 60 хв, а в другому – 35 хв. Таким чином, на веб-сервері від запуску формування прогнозу до отримання результатів користувачем пройшло 167 хв, а аналогічний процес на VM зайняв у 4 рази менше – 42 хв.

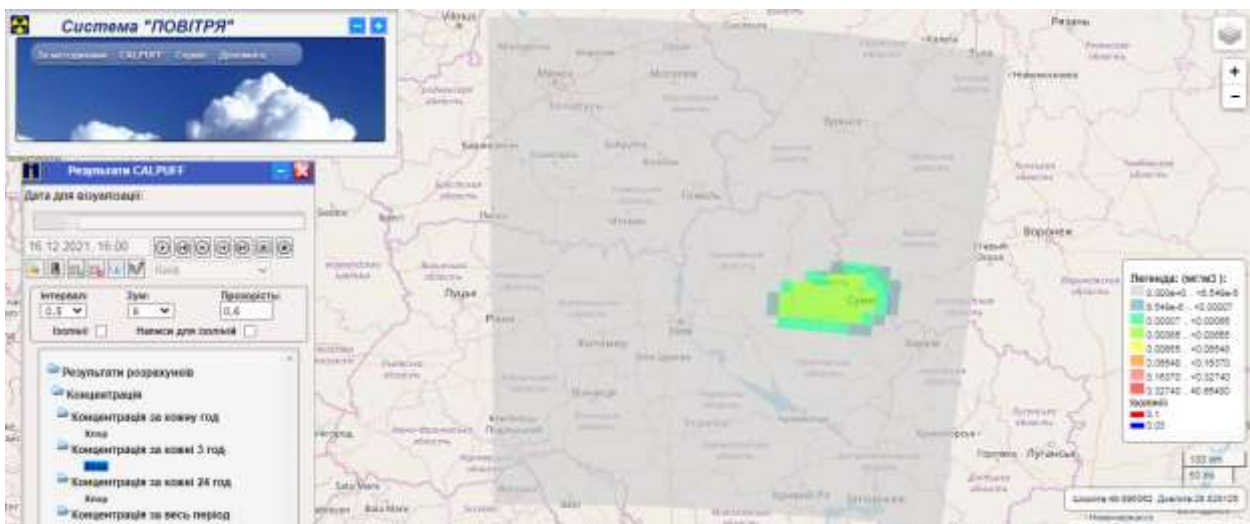


Рисунок 8 – Результат розрахунку системи «Повітря» на тестових даних

#### 5. Висновки

У роботі удосконалено веб-систему прогнозування атмосферних забруднень «Повітря» шляхом збереження налаштованої моделі атмосферного перенесення CALPUFF у вигляді попередньо налаштованого образу на базі ОС Ubuntu та інтеграції у системі програмних засобів автоматичного створення віртуальних машин із попередньо збереженого образу після запиту користувача на розрахунок. У попередній реалізації веб-сервіс «Повітря» передбачав постійну роботу VM, що викликало певні проблеми у роботі системи після збоїв у функціонуванні хмарної інфраструктури УНГ. Для вирішення цих проблем в Українській національній грід інфраструктурі було налаштовано образ на базі операційної системи Ubuntu, разом із необхідними бібліотеками CALPUFF. На базі створеного образу можна швидко запускати VM і розпочинати розрахунок. Після розрахунку система «Повітря» видаляє VM, звільнюючи ресурси хмари для інших задач. Винесення модуля розрахунків в окрему віртуальну машину робить систему більш стійкою до відмов і за наявності доступ-



них ресурсів у хмарі дає можливість запускати декілька ВМ і робити паралельно декілька розрахунків.

На прикладі тестових розрахунків показано, що завдяки винесенню модуля розрахунку атмосферного перенесення у хмарну інфраструктуру вдалося зменшити час розрахунку моделі CALPUFF майже у 2 рази, а час завантаження даних прогнозу погоди зменшено більше, ніж у 10 разів. Сумарно час виконання розрахунку системи після запиту користувача зменшився для тестової конфігурації у 4 рази. Крім того, підвищилась стійкість системи щодо збоїв у роботі хмарної інфраструктури.

Створений архів з операційною системою та налаштованим для роботи CALPUFF також можна переносити за потреби на інші приватні хмари (наприклад, Amazon, Google Cloud Platform, Microsoft Azure) та запускати ВМ на їх базі. Таким чином, розроблена веб-система відповідає сучасним трендам із впровадження хмарних технологій, дозволяє за необхідності масштабування і може бути адаптованою для інших приватних чи публічних обчислювальних хмар.

## СПИСОК ДЖЕРЕЛ

1. Zynovyev M., Svistunov S., Sudakov O., Boyko Y. Ukrainian Grid Infrastructure: Practical Experience. *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. Germany, Dortmund: IEEE, 2007. P. 165–169. doi: 10.1109/IDAACS.2007.4488397.
2. Kovalets I.V., Maistrenko S.Ya., Khalchenkov A.V., Zagreba T.A., Khurtsilava K.V., Anulich S.N., Bespalov V.P., Udovenko O.I. Povitrya web-based software system for operational forecasting of atmospheric pollution after manmade accidents in Ukraine. *Science and Innovation*. 2017. Vol. 13 (6). P. 11–22. doi: 10.15407/scin13.06.013.
3. Kovalets I.V., Maistrenko S.Ya., Khalchenkov O.V., Polonsky O.O., Dontsov-Zagreba T.O., Khurtsilava K.V., Udovenko O.I. Adaptation of the Web-Service of Air Pollution Forecasting for Operation within Cloud Computing Platform of the Ukrainian National Grid Infrastructure. *Science and Innovation*. 2021. Vol. 17 (1). P. 90–100. doi: 10.15407/scine17.01.078.
4. OpenStack – комплекс проєктів вільного програмного забезпечення. URL: <https://www.openstack.org/>.
5. Synkevych R. Openstack-Calpuff. GitHub – Веб-сервіс для хостингу ІТ-проєктів [online]. URL: <https://github.com/Synkevych/openstack-calpuff/>.
6. Scire J.S., Strimaitis D.G., Yamartino R.J. A user's guide for the CALPUFF dispersion model (Version 5). USA, Concord: Earth Tech, Inc., 2000. 521 p.

*Стаття надійшла до редакції 29.03.2022*