КІБЕРНЕТИКА та КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

УДК 004.9 DOI:10.34229/2707-451X.21.3.7

Т.О. БАРДАДИМ, О.В. ЛЕФТЕРОВ, С.П. ОСИПЕНКО

ДОСВІД ТЕСТОВОГО РОЗГОРТАННЯ OPENSTACK І ПОРІВНЯННЯ ВІРТУАЛЬНОГО ТА РЕАЛЬНОГО КЛАСТЕРНИХ СЕРЕДОВИЩ

Наведено інформацію про досвід тестового розгортання та використання хмарного сервісу OpenStack та віртуального кластерного середовища Slurm для забезпечення відтворюваних та масштабованих наукових обчислень із застосуванням сучасних технологічних рішень, розрахованих як для хмарних (OpenStack, AWS, Google), так і для кластерних платформ (Slurm). [1]

Ключові слова: *хмарні технології, відтворювані обчислення, платформа кластерна.*

Вступ. Згідно з планами розвитку Національної академії наук України «… 2013 року було започатковано Цільову комплексну програму наукових досліджень НАН України «Грід-інфраструктура і грід-технології для наукових і науково-прикладних застосувань», в рамках якої планується розвивати хмарні технології, створити об'єднану хмару Українського національного гріду (УНГ) і забезпечити його системне інтегрування в Європейську інфраструктуру» [1]. Мова йде про інтеграцію в Європейську хмару відкритої науки (EOSC) шляхом об'єднання наявних і нових інфраструктур країн учасників [2].

Однією з важливих цілей, яку прагнуть досягнути за допомогою EOSC, побудованої на принципах відкритої науки, – подолати кризу відтворюваності наукових досліджень. «… 2012 року з'явилась інформація про відтворюваність даних біохімічних досліджень, на яких ґрунтувалися результати, що ввійшли до наукових публікацій. Було встановлено, що половина даних не відтворюються і, фактично, є «хижацькими». Інше дослідження показало, що 80 % статей не мають посилання на дані досліджень, до яких потенційно є доступ. Саме ці факти взяли до уваги експерти найвищого рівня, які почали проектувати EOSC. Вони дійшли до висновку, що пов'язані з цими негараздами потенційні втрати бюджету ЄС на дослідження – а це загалом близько 120 млрд. євро – можуть бути значними» [1]. Як повідомляється на сайті Національної хмари відкритої науки (NOSC-UA, див. http://cloud-5.bitp.kiev.ua/?page_id=21) – «Для забезпечення наукової і науково-організаційної діяльності в НАН України в частині впровадження хмарних технологій та в рамках робіт з побудови прототипу Національної хмари відкритої науки України як складової частини Європейської хмари відкритої науки, в 2018 році в Інституті теоретичної фізики ім. М.М. Боголюбова НАН України побудована сучасна хмарна інфраструктура під управлінням програмної системи OpenStack з використанням 12 серверів».

[1] За підтримки Національної академії наук України (тема ВФ.115.41).

Програмна система з відкритим кодом OpenStack [3] широко використовується у світі для створення публічних та приватних хмарних сервісів, зокрема наукових. Найбільш відомим науковим хмарним сервісом, побудованим з використанням OpenStack, є CERN OpenStack Cloud [4, 5]. Цей хмарний науковий сервіс нараховує понад 300 тис. обчислювальних ядер та об'єднує тисячі науковців у всьому світі[2].

Склад і призначення базових компонент хмарного сервісу OpenStack. Проект з розробки хмарної операційної системи OpenStack з'явився у 2010 році як спільний проект NASA та Rackspace. До його складу входили усього дві підсистеми – Nova для запуску віртуальних серверів, Swift для зберігання даних. На сьогодні OpenStack розвивається під керівництвом OpenStack Foundation. До складу розробників входить більше ніж 70 тис. індивідуальних членів та більше ніж 600 корпоративних. Нині OpenStack складається з декількох десятків підсистем, а його код складає більше 20 мільйонів рядків. Мова програмування, якою написана основна частина коду – Python. Цей код розповсюджується за ліцензією Apache 2.0. Серед компаній, які мають найбільшу кількість інсталяцій – HP, IBM, Oracle, Red Hat, SUSE [6].

Архітектура OpenStack складається з множини підсистем, кожна з яких розробляється окремо. Конкретна інсталяція хмарної операційної системи OpenStack може містити усього декілька підсистем. Найбільш базова підсистема – OpenStack Compute (Nova). Вона дозволяє використовувати віртуальні машини та контейнери Docker[3] (створювати, запускати, робити «знімки» стану віртуальних машин та інше). Наступна підсистема – OpenStack Networking (Neutron), яка відповідає за віртуальну мережу, що з'єднує віртуальні машини та віртуальне/реальне мережеве обладнання. Ця підсистема дозволяє створювати віртуальні – мережі, мережеві порти, маршрутизатори, «плаваючі IP адреси». Останні дозволяють отримати віртуальним машинам зовнішні IP адреси для доступу до глобальної мережі. Підсистема ідентифікації OpenStack Keystone є централізованим каталогом користувачів та сервісів, до яких їм надано доступ. OpenStack Image Service (Glance) містить каталог образів віртуальних машин, які використовуються для запуска їхніх екземплярів. OpenStack Dashboard (Horizon) – надає користувачу веб-доступ до системи. Існує багато інших підсистем, з повним їх переліком для поточної версії «Wallaby» можна ознайомитися на сайті OpenStack[4] [6].

Тестове розгортання хмарного сервісу OpenStack та віртуального кластерного середовища Slurm[5]. *Мета* тестового розгортання – створити масштабоване обчислювальне середовище (ОС) для проведення відтворюваних наукових обчислень із використанням сучасних технологічних рішень, яке може бути застосовано як на хмарних (OpenStack, AWS, Google), так і на реальних кластерних платформах (Slurm).

Наявність такого середовища дає змогу розробляти відтворювані схеми аналізу наукових даних з використанням мов опису схем на кшталт Common Workflow Language (CWL) або Workflow Description Language (WDL), які можуть бути застосовані на різноманітних хмарних та кластерних платформах [7].

Зокрема, в задачах класифікації біомедичних даних виникає потреба у перевірці надійності отриманої класифікаційної моделі шляхом застосування техніки ресамплінгу (наприклад, кросвалідації), яка з одного боку – трудомістка (інколи кількість повторів сягає декількох сотень), а з іншого – добре піддається паралельному одночасному обрахунку за наявності ресурсів.

[2] Див. https://www.openstack.org/videos/summits/virtual/10-years-of-OpenStack-at-CERN-From-0-to-300k-cores
[3] Використання технології Docker потребує окремих налаштувань.
[4] Див. https://releases.openstack.org/wallaby/
[5] Менеджер ресурсів для кластерних середовищ. Див. https://en.wikipedia.org/wiki/Slurm_Workload_Manager

Наприклад, пошук інформативних ознак з використанням методів негладкої оптимізації [8–12] на масиві даних генної експресії великої розмірності на звичайному комп'ютері триває більше десяти годин. Багаторазове застосування цього метода на різних підмножинах даних можливе в умовах наявності як достатніх обчислювальних ресурсів, так і технологічних рішень, які забезпечують відтворюваність отриманих результатів. Ще одним прикладом є застосування методів побудови ансамблів класифікаторів, гібридного підходу для їх побудови [13, 14]. Моделі, які будуються різними методами, можуть також створюватися паралельно, але треба мати можливість забезпечити усі умови, які здатні підтримувати відтворюваність таких розрахунків на різних платформах. Представлене у цій роботі тестове обчислювальне середовище використовує сучасні технології, які дозволяють досягти цієї мети.

Тестове середовище представляє собою віртуальний кластер як сукупність віртуальних машин, об'єднаних у віртуальну мережу, на яких встановлено відповідне програмне забезпечення, включаючи менеджер ресурсів Slurm. Кластер складається з 4-х обчислювальних та одного керуючого вузла. Мережева топологія віртуально кластера показана на рис. 1. До речі, ця топологія відтворюється у веб інтерфейсі OpenStack автоматично.

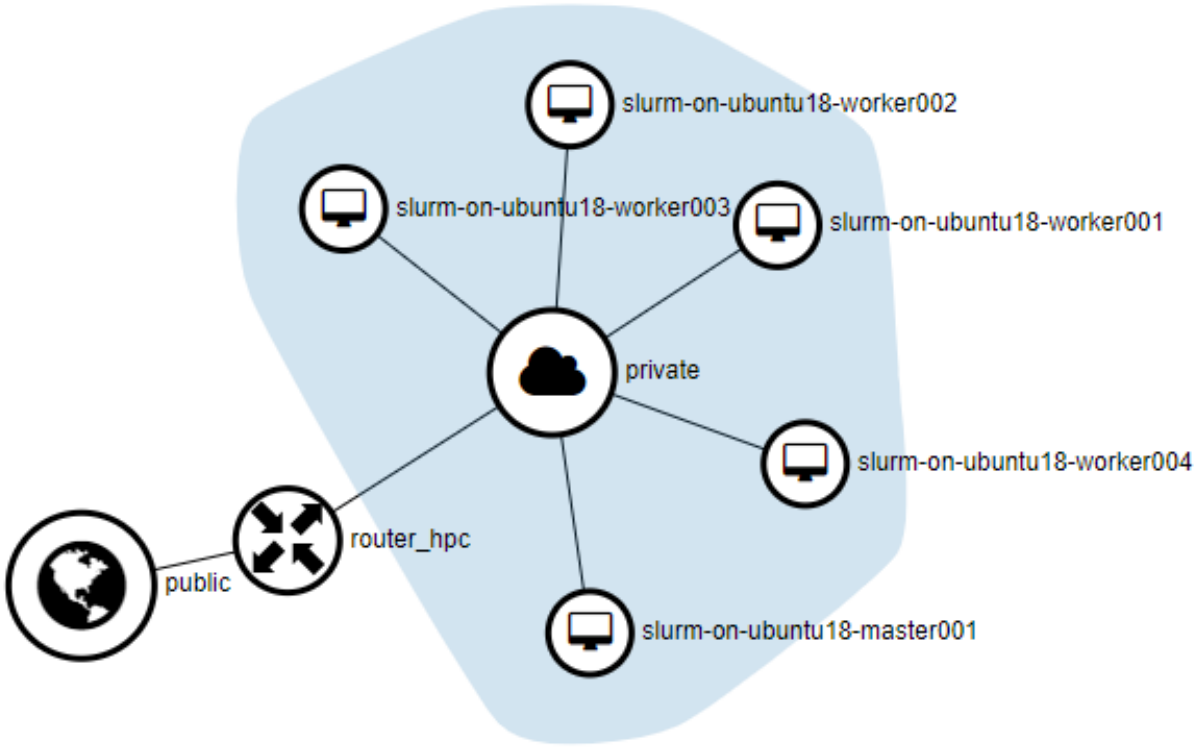

РИС. 1. Мережева топологія віртуального кластера

Для проведення тестових обчислювальних експериментів було розгорнуто віртуальний кластер (під управлінням менеджера ресурсів Slurm) у хмарному середовищі OpenStack. Існує декілька способів встановити середовище OpenStack:

- безпосередньо з доступного первинного коду,
- використовуючи наявні дистрибутиви,
- за допомогою інструментів його розгортання.

Для тестового використання найбільш простим є спосіб розгортання за допомогою пакета DevStack[6]. Цей спосіб дозволяє встановити OpenStack на персональному комп'ютері і навіть на віртуальній машині за допомогою декількох команд.

Отримане за допомогою пакету DevStack середовище дозволяє використовувати основні функції OpenStack. Наприклад, у цьому середовищі можна розгорнути віртуальний кластер Slurm за допомогою інструмента автоматизації розгортання ElastiCluster, що і було нами зроблено.

До обчислювального середовища увійшли наступне обладнання та програмні засоби з відкритим програмним кодом:

- сервер Dell PowerEdge T710, на якому було розгорнуто обчислювальне середовище, оснащений двома процесорами Intel Xeon X5650 2.67 GHz, 24 Gib оперативної пам'яті DDR3, RAID контролером, масивом RAID_1 з двох 2,5" дисків WD VelociRaptor 160 GB 10K RPM (для розміщення основної операційної системи) та логічним томом з двох SSD дисків (фізичних томів) на 240 та 480 Gb (для розміщення віртуальних машин);
- операційна система GNU/Linux – Ubuntu server amd64 (версія 18.04.5 для віртуальних машин та 20.04.2 для базової операційної системи);
- OpenStack (версія XENA) – хмарна операційна система, встановлена на сервері Ubuntu 20.04.2. Версія XENA (знаходиться в стані розробки) розгортається за замовченням пакетом DevStack[7];
- ElastiCluster[8] (версія 1.3.dev28) – програмне забезпечення, розроблене центром інформаційних технологій Цюріхського університету (Service and Support for Science IT) для розгортання на хмарних платформах (OpenStack, AWS, Google, Azure) кластерів (Slurm, Kubernetes, Hadoop+Spark, Ceph та інших) – встановлено на окрему віртуальну машину "ElastiCluster" на OpenStack;
- Ansible[9] (версія 2.9.24) – програмне забезпечення для автоматизованого розгортання програмних додатків у хмарних, кластерних середовищах – встановлено на віртуальну машину "ElastiCluster" на OpenStack під час інсталяції програмного забезпечення ElastiCluster;
- Slurm (версія 17.11.2) – менеджер ресурсів для кластера – встановлено на 5 віртуальних машинах віртуального кластера (див. рис. 1) на OpenStack за допомогою ElastiCluster;
- OpenMPI (версія 2.1.1) – програмне забезпечення для передачі сповіщень між процесами паралельних обчислень – встановлено на 4 віртуальних машинах (обчислювальних вузлах "worker[001-004] ") віртуального кластера на OpenStack за допомогою ElastiCluster;
- Singularity[10] (версія 3.7.1) – програмне забезпечення для контейнеризації додатків, що використовуються на кластерах – встановлено на 4 обчислювальних вузлах "worker[001-004]" віртуального кластера на OpenStack за допомогою Ansible;
- Toil[11] (версія 5.4.0) – програмне забезпечення для проведення масштабованих, портабельних (які можуть бути застосовані на хмарах, кластерах, персональних комп'ютерах), відтворюваних наукових обчислень з використанням схем аналізу, створених мовами CWL або WDL, яке розробляється лабораторією обчислювальної геноміки Інституту геноміки у складі Університету Каліфорнії у Санта-Крус – встановлено на керуючій вузол "master001" віртуального кластера на OpenStack.

[6] Див. https://docs.openstack.org/devstack/latest/
[7] Слід зауважити, що також існує можливість обирати з поміж 10-ти інших, стабільних версій.
[8] Див. https://github.com/elasticluster/elasticluster
[9] Див. https://www.ansible.com/
[10] Див. https://sylabs.io/singularity/
[11] Див. http://toil.ucsc-cgl.org/

Послідовність дій для розгортання обчислювального середовища:

- встановити операційну систему Linux на основний комп'ютер (сервер);
- розгорнути за допомогою DevStack потрібну версію OpenStack;
- засобами OpenStack створити та налаштувати проект (створити потрібні віртуальні мережі, маршрутизатори, «плаваючі адреси», групи безпеки, ключі доступу, користувачів, надати їм права доступу та інше);
- засобами OpenStack у створеному проекті підготувати образ віртуальної машини Linux, за допомогою якого будуть розгорнуті вузли віртуального кластера, та налаштувати віртуальну машину (встановити версію Python 2.7 – для роботи Ansible, надати права безпарольного доступу до команди «sudo» користувачу, від імені якого буде здійснено розгортання віртуального кластера та інше);
- засобами OpenStack запустити та налаштувати віртуальну машину, на якій розгорнути програмний засіб ElastiCluster;
- з віртуальної машини, на якій встановлено ElastiCluster, налаштувати файл параметрів конфігурації віртуального кластеру Slurm та розгорнути його, перевірити його працездатність;
- з віртуальної машині з ElastiCluster налаштувати параметри файлу конфігурації Ansible для розгортання Singularity та виконати її розгортання, перевірити працездатність на робочих вузлах віртуального кластера;
- на керуючому вузлі віртуального кластера у віртуальному середовищі Python встановити Toil та перевірити його працездатність.

Середовище готове для використання.

Тестове дослідження та порівняння віртуальних та реальних кластерних обчислювальних середовищ. *Мета* тестового використання – перевірка працездатності розгорнутого віртуального кластерного середовища, реального кластерного середовища та досягнення портабельності, масштабованості й відтворюваності наукових обчислень.

Перевірка портабельності та масштабованості. Портабельність передбачає можливість здійснювати відтворювані обчислення на різних кластерних платформах (як реальних, так і віртуальних) без необхідності внесення суттєвих змін як у налаштування самих платформ, так і у програмний код, пов'язаний з обчисленнями. Нами успішно перевірена можливість перенесення обчислень з віртуального кластера (під керуванням менеджера ресурсів Slurm) в OpenStack на реальний, «класичний» кластер Інституту кібернетики імені В.М. Глушкова НАН України, а саме на вузол n4202.icyb кластера СКІТ-4.5.

Перевірка масштабованості мала на меті з'ясувати, за яких значень параметрів масштабування можливо оптимально використовувати доступні обчислювальні ресурси (зменшити час розрахунку та навантаження на обчислювальну інфраструктуру, а саме – кількість задіяних процесорних ядер, та залученої оперативної пам'яті) як у віртуальному кластері Slurm, розгорнутому у тестовому хмарному середовищі OpenStack, так і на СКІТ-4.5. Підбір параметрів масштабування набуває важливості для довготривалих обчислень (біомедичних, зокрема), особливо якщо ці розрахунки відбуваються на орендованих (платних) хмарних платформах.

Для перевірки масштабованості була обрана задача крос-валідації моделі лінійної класифікації, побудованої з використанням негладкої оптимізації (із застосуванням модуля NonSmoothLC, який був розроблений Ю.П. Лаптіним) [8–12, 14, 15]. Саме крос-валідація обрана тому, що її проведення на біомедичних даних вимагає залучення великої кількості обчислювальних ресурсів, а також дозволяє застосувати модель масштабування, де окремі завдання можуть виконуватися незалежно одне від одного.

Крос-валідація класифікаційної моделі була проведена на біомедичних даних з експресії генів[12]. До моделі увійшли 249 найбільш інформативних показників генної експресії, попередньо відібраних з множини 20 тис. Кількість спостережень дорівнювала 152. Обчислення здійснювалися у програмному середовищі R (версія 4.1.1) з усіма необхідними для проведення крос-валідації пакетами (включно з NonSmoothLC), розташованими у контейнеризованому додатку Singularity. Контейнеризація позбавила необхідності інсталяції середовища R та його бібліотек безпосередньо на кластері (у разі використання реального кластера для звичайного користувача це було б неможливо). Крім того, контейнеризація забезпечила одну з умов портабельності обчислень між віртуальним та реальним кластерними середовищами. Дані та програмний код крос-валідації було розташовано поза контейнером, що дало змогу використовувати адаптовану до кластерних обчислень версію коду.

Для забезпечення масштабування обчислень залучено технологію MPI. Ця технологія використовувалася для розподілу завдань між паралельними процесами (контейнерами із R) із використанням мови Python. Додаток на Python запускався як множина паралельних процесів, які в свою чергу запускали екземпляри контейнерів для виконання обрахунків, передаючи їм як параметри індекси наборів даних, на яких необхідно було провести крос-валідацію, та іншу необхідну інформацію. Обчислення проводилися за допомогою контейнеризованих додатків. Результати аналізу зберігались у вигляді файлів даних R. Агрегація отриманих результатів крос-валідації проводилася окремим не паралельним процесом.

Для тестування масштабованості обрано крос-валідацію на основі методу Монте Карло (leave-group-out cross-validation) з 36-ма повторами. Число 36 обране тому, що вузли кластера СКІТ-4.5 містять саме стільки обчислювальних ядер, і важливо було перевірити ефективність їх одночасного використання для цієї задачі. Тестування проводилося лише на одному вузлі кластера СКІТ-4.5 для порівняння з результатами тестування на віртуальному кластері OpenStack, розташованому також лише на одному сервері.

Масштабування проводилося наступним чином: відбувалося покрокове збільшення кількості паралельних процесів для проведення крос-валідації та фіксувався час її виконання, зберігалася динаміка навантаження на вузол та використання оперативної пам'яті[13]. Облік вказаних показників відбувався автоматично із використанням скриптів, які керували процесом виконання обчислювальних завдань. Було проведено 5 серій таких тестувань для віртуального та класичного кластерів. Серії були необхідні для оцінки варіативності вказаних величин. Для класичного кластера одна серія тривала більше 4-х годин та містила 36*36 = 1296 результатів крос-валідації (проведеної за різних значень одночасно працюючих процесів). Результати зберігалися у вигляді файлів даних R, які потім були агреговані, та текстових файлів, які містили протоколи динаміки змін показників, отриманих під час обчислень.

Для віртуального кластера доступними були лише 4 віртуальні вузли, на кожному з яких було доступно 4 Gb пам'яті. Це обмежило кількість одночасно працюючих контейнеризованих процесів до 4 для усього кластера. Тривалість обчислень для однієї серії була меншою за годину і кількість результуючих файлів дорівнювала 4*36 = 144.

[12] Ці дані є підмножиною набору даних The Cancer Genome Atlas (TCGA) та отримані за посиланням http://download.cbioportal.org/gbm_tcga_pub2013.tar.gz у 2019 році.

[13] Динаміка зміни показників навантаження та використання пам'яті фіксувалася шляхом заміру цих показників за допомогою програми top (входить до складу Linux) з 10 секундним інтервалом. У якості показника навантаження обрано середнє за хвилину навантаження (loads average), який вимірюється програмою top.

На рис. 2 показано деякі результати порівняння масштабованості для задачі крос-валідації для обчислювальних середовищ OpenStack та СКІТ-4.5. Окремими маркерами на рисунку позначені спостереження швидкості, отримані у 5-ти серіях обчислень для різних значень кількості паралельних процесів. Також на рисунку показані регресійні лінії для обох множин даних, що належать порівнюваним кластерним платформам. Ці регресійні лінії (моделі) отримані шляхом проведення нелінійного регресійного аналізу з використанням пакета "drc" («Dose-Response Curves») версії 3.0-1 у середовищі R [16, 17]. Для побудови моделі обрано функцію «LL2.4» як параметр функції "drm".

Порівнюючи регресійні лінії варто зауважити, що для платформи СКІТ-4.5 ця лінія має виражений нелінійний характер. Регресійна лінія для OpenStack має лінійний характер. Тобто, для віртуального кластера в OpenStack приріст швидкості відбувається лінійно зі збільшенням кількості паралельних процесів, принаймні це справедливо до 4-х паралельних процесів включно (для перевірки лінійності на більшій кількості процесів на OpenStack необхідно додатково налаштувати параметри використання оперативної пам’яті у Slurm). На відміну від OpenStack, для СКІТ-4.5 спостерігається відсутність приросту швидкості при переході від одного до двох паралельних процесів, та при додаванні процесів після 4-х. Можливо, ситуація мала б інший вигляд, якщо було б збільшено кількість вузлів СКІТ-4.5, а не кількість процесів на одному вузлі (як це було зроблено з віртуальними вузлами на OpenStack). Але важливо було максимально урівняти умови (кількість фізичних серверів, зокрема) порівняння обчислювальних середовищ.

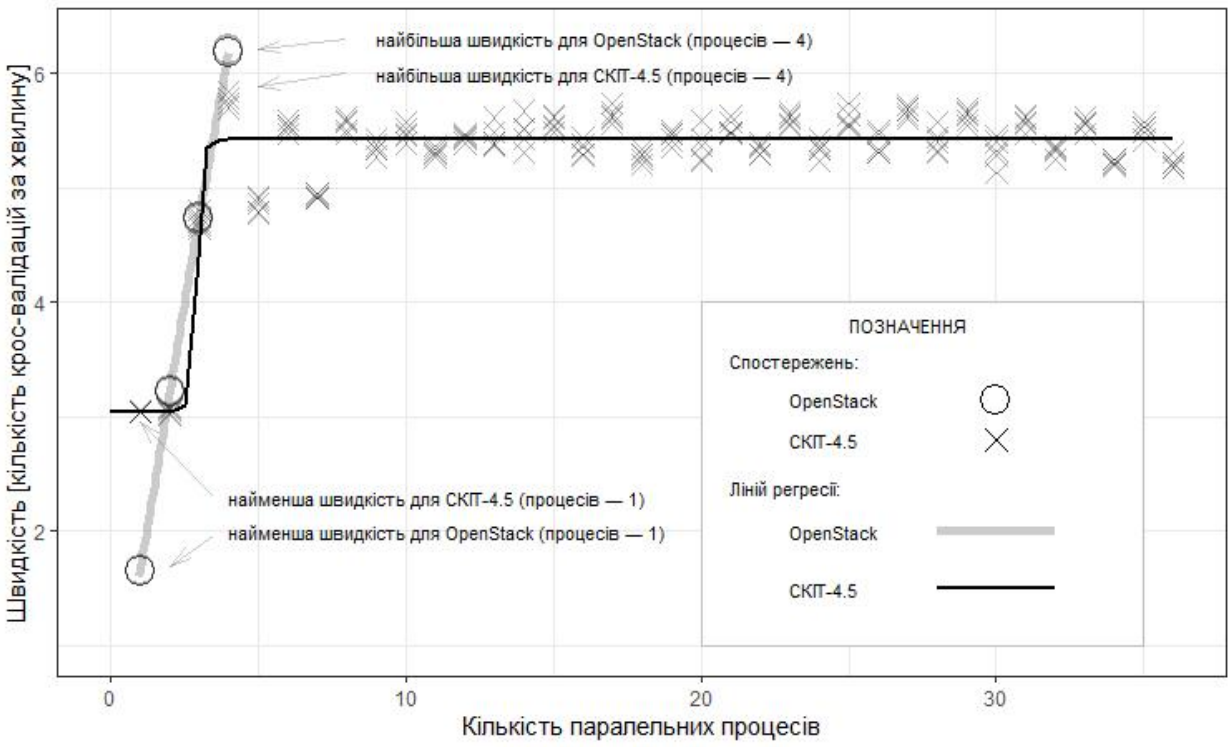

РИС. 2. Порівняльний розподіл та лінії регресії швидкості обчислень у залежності від кількості паралельних процесів для двох кластерних середовищ – СКІТ-4.5 (n = 180) та OpenStack (n = 20) (кількість серій обчислень = 5)

Аналізуючи рис. 2 також можна зауважити, що коли обчислення відбувалися із використанням лише одного процесу, швидкість СКІТ-4.5 майже у двічі перевищувала швидкість OpenStack. Це можна пояснити тим, що на порівнюваних кластерах фактично була задіяна різна кількість процесорних ядер[14], попри те, що спеціальним параметром[15] було вказано використовувати лише одне ядро на одну задачу. Також це можна пояснити тим, що на СКІТ-4.5 встановлено більш потужні процесори (Intel Xeon E5-2695 проти Intel Xeon X5650 на OpenStack), які під час запуску тестових завдань працювали не у віртуальному (всередині віртуальних машин), а в реальному режимі[16] (див. рис. 2, позначка «найменша швидкість …»). Та коли кількість паралельних процесорів досягла 4-х, – перевагу у швидкості вже отримував OpenStack (див. рис. 2, позначка «найбільша швидкість …»). Причини такої різниці у швидкості нами остаточно не з'ясовані. Можливо, вони пов'язані з особливостями налаштуванням параметрів обчислень на кластерних середовищах. На підтвердження цього припущення можна навести той факт, що при використанні іншого вузла СКІТ-4.5[17] для 36 паралельних процесів отримано швидкість 8.2 крос-валідацій на хвилину проти 5.3 для вузла, на якому проводилися основні обчислення.

Порівняння навантаження на сервер для двох кластерів передбачало реєстрацію показників навантаження та кількості використаної пам'яті. Моніторинг показників відбувався неперервно з інтервалом у 10 сек. для серії обчислень у цілому. Зважаючи на те, що вузли кластера OpenStack були віртуальними, відбувалася реєстрація навантаження на сам фізичний сервер. Такий же моніторинг проводився і для вузла кластера СКІТ-4.5. Різниця полягала лише у тому, що програму моніторингу доводилося час від часу перезапускати (в автоматичному режимі) на сервері СКІТ-4.5, тому що її виконання через певний час переривалося самим сервером. Це пояснює, чому лінії, які відображають динаміку змін у показниках, мають переривчастий характер для СКІТ-4.5 (див. рис. 3). На рис. 3 з метою порівняння кластерних платформ показано моніторинг однієї повної серії для OpenStack (4 завдання) та частину серії для СКІТ-4.5 (перших 4 завдання з 36 у серії).

Порівняємо характер кривих досліджуваних величин для обох кластерів. Для OpenStack діапазон навантажень змінюється у межах ≈ 0.6 – 3.7 умовних одиниць навантаження та має більш пологий характер ніж у СКІТ-4.5. Зважаючи на те, що на сервері OpenStack розташована сама хмарна система, можна зробити припущення, що процеси, які належать OpenStack, впливали на те, що навантаження не зменшувалося після закінчення завдання (наприклад, відбувалося управління кешем). Для сервера СКІТ-4.5 границі 4-х завдань легко розгледіти тому, що в цих місцях зафіксовано різке зменшення навантаження. Діапазон навантажень для СКІТ-4.5 коливається у межах ≈ 0 – 67 умовних одиниць. Спробуємо пояснити таку різницю у діапазонах навантаження. Як зазначається у керівництві до програми top: «У багатоядерній системі спочатку слід поділити середнє значення навантаження на кількість ядер процесора, щоб отримати подібний показник» [18].

[14] Для однієї задачі на СКІТ-4.5 отримано значення 1800 % (що, є еквівалентом повного використання 18 ядер) у колонці «% CPU» таблиці, яка сформована під час моніторингу використання ресурсів за допомогою програми top. А для однієї задачі на OpenStack отримано значення 100 % (1 використане ядро).

[15] Параметр налаштування «--cpus-per-task» для програми sbatch, яка створює завдання Slurm, було встановлено рівним 1 для обох порівнюваних кластерів.

[16] Порівняння процесорів див. на http://cpuboss.com/cpus/Intel-Xeon-X5650-vs-Intel-Xeon-E5-2695-v3

[17] Йдеться про вузол n4203.icyb, на якому була запущена задача для 36 паралельних процесів без використання Slurm, але тільки із використанням MPI.

[18] A Guide to the Linux "Top" Command https://www.booleanworld.com/guide-linux-top-command/

Тоді максимальне значення для СКІТ-4.5 буде 67/36 = 1,86 (де 36 – сумарна кількість ядер), а для OpenStack – 3,7/4 = 0,92. У знаменнику вказано 4 тому, що для OpenStack обчислення відбуваються за участі 4-х віртуальних машин, на кожній з яких із доступних 3-х віртуальних ядер використовувалося лише одне[19]. Тоді навантаження на вузол СКІТ-4.5 було у двічі більшим (1,86/0,92 ≈ 2) ніж на OpenStack під час виконання 4-х паралельних процесів.

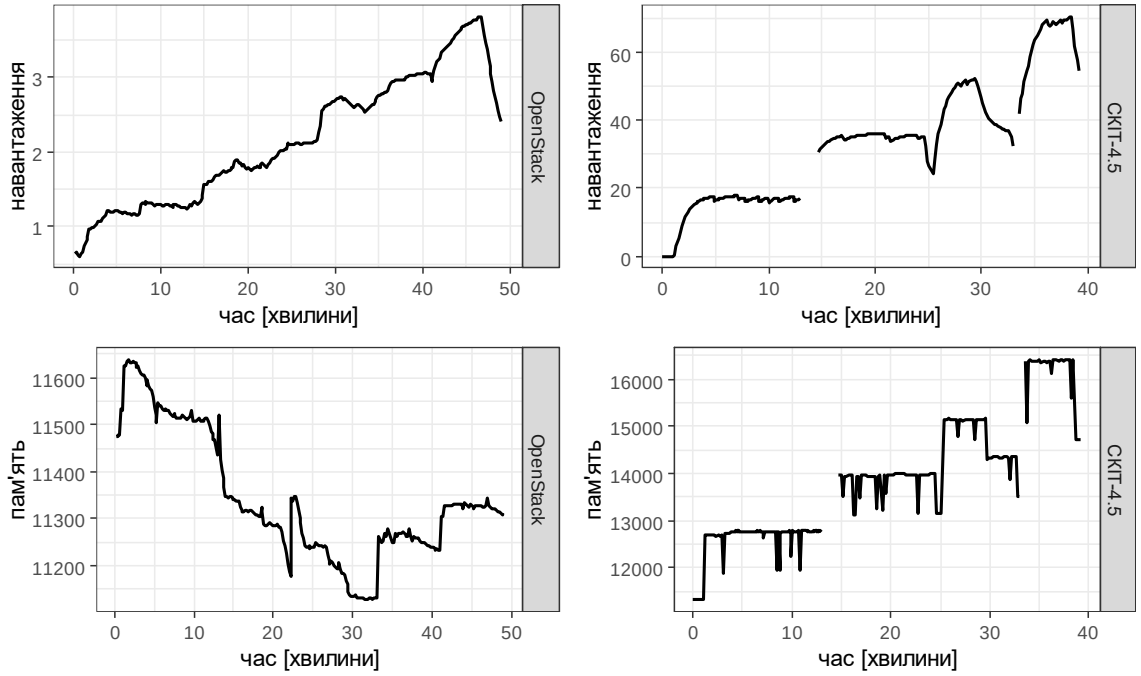

РИС. 3. Порівняння навантаження та використання оперативної пам’яті для серверів OpenStack та СКІТ-4.5 при послідовному виконанні 4-х крос-валідацій із збільшенням кількості паралельних процесів з 1 до 4

Розглянемо іншу групу графіків, які зображають динаміку змін у використанні оперативної пам’яті. Невелике зменшення використаної оперативної пам’яті для OpenStack під час збільшення навантаження на сервер можна пояснити тим, що по-перше – пам’ять, яка використовувалася для виконання завдань, була виділена 4-м віртуальним машинам заздалегідь, і тому зміни у її використанні суттєво не позначилися на кількості вільної пам’яті на самому сервері, по-друге – на цьому ж сервері встановлено і сам OpenStack, який, можливо, керуючи власними процесами, змінював кількість вільної пам’яті. Попри те, на графіку використання пам’яті для OpenStack можна побачити короткі ділянки, на яких відбувалося різке зростання використання пам’яті (на відмітках 2, 22, 34 та 42 хвилини). Ці ділянки графіка відповідають початкам 4 завдань з різною кількістю паралельних процесів. Зважаючи на те, що 4-ма віртуальними машинами було використано 1,25 * 4 = 5 Gb пам’яті, та на вузлі СКІТ-4.5 16,3 – 11,3 = 5 Gb, можна вважати, що використання пам’яті на цих різновидах кластерів є однакове за обсягом.

[19] У разі віртуального кластера OpenStack параметр налаштування «--cpus-per-task» для програми sbatch зміг обмежити до одиниці кількість ядер на процес. На вузлі n4202.icyb розділу кластера СКІТ-4.5, очевидно, цей параметр був проігнорований.

Перевірка відтворюваності обчислень. Для наукових обчислень відтворюваність має особливе значення, тому ми включили її перевірку в наше тестове використання OpenStack. Під відтворюваністю у контексті задачі крос-валідації розуміється можливість отримати такі самі результати як при зміні параметрів масштабування, так і при проведенні обчислень на різних платформах.

Перевірка була виконана наступним чином. Результатом крос-валідації є показники: надійності отриманої моделі (коефіцієнт F1 Соренсена) [18], величина проміжку між класами та коефіцієнти отриманої моделі (249 коеф. моделі + 1 вільний член = 250). Тобто, в цілому 252 змінні, які перевірялися на вітворюваність для 36 (варіантів даних) * 36 (варіантів кількості паралельних процесів) * 5 (серій) = 6480 варіантів обчислень у випадку СКІТ-4.5 та 36*4*5 = 720 варіантів для OpenStack. Фактично перевірялася матриця результатів крос-валідації розміром 252 стовпчиків
на 7200 рядків. Процедура аналізу цієї матриці полягала у підрахунку стандартних відхилень для кожного із 252 показників по кожному з 36-ти варіантів даних, використаних при крос-валідації (як для кожного середовища окремо, так і для двох середовищ разом). У разі абсолютного співпадіння результатів сумарне стандартне відхилення дорівнювало б нулю. В реальності ми отримали середнє значення стандартного відхилення для змінної рівним 3.0e-07. Максимальне стандартне відхилення дорівнювало 7.5e-05. Тобто, на відтворюваність розрахунків не вплинуло ні масштабування, ні перенесення їх в інше кластерне середовище.

Відтворюваність наукових обчислень забезпечується не тільки тим, що вдається отримати один і той самий результат у складних розрахунках, у різних обчислювальних середовищах, але й тим, що забезпечується фіксація, протоколювання вхідних параметрів та даних. Для досягнення цієї більш широкої мети застосовуються мови опису відтворюваних схем аналізу та інструменти для запуску таких схем. Зважаючи на те, що інструмент для використання схем аналізу toil вдалося розгорнути тільки на віртуальному кластері OpenStack, для збереження портабельності додатку між кластерними середовищами ми не використовували на цьому етапі підхід із застосуванням схем аналізу на відповідних мовах.

Переваги використання OpenStack для проведення сучасних наукових обчислень. Проведене розгортання та тестування хмарної платформи OpenStack дозволило переконатися у наступних перевагах застосування хмарних обчислювальних середовищ у порівнянні із десктопними або тільки кластерними:

гнучкість – для хмарних сервісів AWS, Google, OpenStack доступні інструменти автоматичного розгортання складних обчислювальних середовищ для аналізу великих наукових даних таких як віртуальні кластери Slurm, Hadoop+Spark, Kubernetes. У цій роботі ми тестували лише віртуальний кластер Slurm у порівнянні з аналогічним «класичним» кластером. Завдяки гнучкості розгортання можна динамічно збільшувати кількість вузлів для забезпечення більшої масштабованості;

портабельність – наукові додатки, підготовлені для використання на «класичному» кластері, можуть бути перенесені на хмарні віртуальні кластери (перевірено для OpenStack) за умови, що вони мають модель паралельності, де окремі завдання можуть виконуватися незалежно одне від одного та можлива їх контейнеризація із використанням технології Singularity. У нашому тестуванні показана перевага у швидкості розрахунку на такій моделі паралельності для задачі крос-валідації. Якщо буде інтенсивно використовуватися передача повідомлень технологією MPI, можна передбачити переваги «класичного» кластера тому, що там застосовуються спеціальні високошвидкісні канали передачі повідомлень;

перспективність – зважаючи на те, що в планах розвитку НАН зафіксовано інтеграцію в Європейську хмару відкритої науки, а в Україні вже існує прототип Національної хмари відкритої науки, побудованої на основі платформи OpenStack, представляється перспективною розробка портабельних біомедичних додатків, які б можна було використовувати і в цьому середовищі.

Результати. Проведено успішне тестове розгортання хмарної платформи OpenStack та віртуального кластера Slurm. Створено тестовий біомедичний додаток, який використовує сучасні програмні засоби. За допомогою цього додатку перевірено його портабельність між хмарним віртуальним кластером OpenStack та «класичним» кластером без втрати відтворюваності обчислень. Показані деякі переваги використання саме віртуального кластера, які полягають у досягненні більшої швидкості розрахунків для окремих задач.

Подяка. Автори щиро вдячні В.М. Горбачуку за наполегливу підтримку при підготовці статті.

Список літератури

1. Загородній А.Г. Європейська хмара відкритої науки як глобальний інструмент наукових досліджень. *Газета «Світ»*. 2020. № 25–26. С. 1–3.
2. Горбачук В., Гавриленко С., Дунаєвський М. До участі України в Європейській хмарі відкритої науки. *Global and Regional Problems of Informatization in Society and Nature Using*. 2021. P. 169–171.
3. Sefraoui O., Aissaoui M., Eleuldj M. OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications.* 2012. **55** (3). P. 38–42.
4. Bell T., Bompastor B., Bukowiec S., Leon J.C., Denis M., van Eldik J., Lobo M.F., Alvarez L.F., Rodriguez D.F., Marino A. Scaling the CERN OpenStack cloud. *Journal of Physics: Conference Series.* **664.** IOP Publishing 2015. P. 022003.
5. Andrade P., Bell T., Van Eldik J., McCance G., Panzer-Steindel B., dos Santos M.C., Traylen S., Schwickerath U. Review of CERN data centre infrastructure. *Journal of Physics: Conference Series.* **396**. IOP Publishing. 2012. P. 042002.
6. Маркелов А. OpenStack. Практическое знакомство с облачной операционной системой. М.: ДМК Пресс, 2016.
7. Strozzi F., Janssen R., Wurmus R., Crusoe M.R., Githinji G., Di Tommaso P., Belhachemi D., Möller S., Smant G., de Ligt J. Scalable workflows and reproducible data analysis for genomics. *Evolutionary Genomics*. Springer, 2019. P. 723–745.
8. Shor N.Z. Nondifferentiable Optimization and Polynomial Problems. London: Kluwer Acad. Publ, 1998. 381 p. https://doi.org/10.1007/978-1-4757-6015-6
9. Шор Н.З., Журбенко Н.Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов. *Кибернетика*. 1971. 3. С. 51–59. https://doi.org/10.1007/BF01070454
10. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. К.: Наук. думка, 1979. 199 с.
11. Лаптин Ю.П., Бардадым Т.А. Проблемы определения коэффициентов точных штрафных функций. *Кибернетика и системный анализ.* 2019. 3. С. 64–79. https://doi.org/10.1007/s10559-019-00147-2
12. Zhuravlev Y.I., Laptin Y.P., Vinogradov A.P., Zhurbenko N.G., Lykhovyd O.P., Berezovskyi O.A. Linear classifiers and selection of informative features. *Pattern Recognition and Image Analysis*. 2017. **27** (3). P. 426–432. https://doi.org/10.1134/S1054661817030336
13. Новоселова Н.А., Том И.Э. Алгоритм ранжирования признаков для обнаружения биомаркеров в данных генной экспрессии. *Искусственный интеллект*. 2013. 3. С. 58–68.
14. Новоселова Н.А., Скобцов В.Ю., Том І.Е., Бардадим Т.О., Горбачук В.М., Осипенко С.П. Сучасні можливості розробки та організації інтелектуальних аналітичних систем. Матеріали IX Всеукраїнської науково-практичної конференції*: «Глушковські читання»,* 18 грудня 2020 р. Київ: М-во освіти і науки України, Київський нац. ун-т імені Тараса Шевченка. 2020. С. 113–116.
15. Bardadym T.O., Gorbachuk V.M., Novoselova N.A., Osypenko S.P., Skobtsov V.Yu., Intelligent analytical system as a tool to ensure the reproducibility of biomedical calculations. *Artificial Intelligence*. 2020. 3. P. 65–78.
16. Ritz C., Strebig J.C., Ritz M.C. Package ‘drc’. Creative Commons: Mountain View, CA, USA. 2016.
17. DeLean A., Munson P., Rodbard D. Simultaneous analysis of families of sigmoidal curves: application to bioassay, radioligand assay, and physiological dose-response curves. *American Journal of Physiology-Endocrinology and Metabolism*. 1978. **235** (2). P. E97.
18. Sørensen T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*. 1948. **5**. P. 1–34.

Одержано 14.09.2021

Бардадим Тамара Олексіївна,
кандидат фізико-математичних наук, старший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України, Київ,
Tamara.Bardadym@gmail.com

Лефтеров Олександр Володимирович,
науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України, Київ,

Осипенко Сергій Петрович,
інженер-програміст
Інституту кібернетики імені В.М. Глушкова НАН України, Київ.
baston888@gmail.com

UDC 004.09

Tamara Bardadym, Oleksandr Lefterov, Sergiy Osypenko

Experience of OpenStack Test Deployment and Comparison of Virtual and Real Cluster Environments

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv
Correspondence: Tamara.Bardadym@gmail.com

Introduction. A brief overview of the properties and architecture of one of the components of the National Cloud of Open Science prototype – the cloud platform OpenStack is given. The list of software and hardware components of the OpenStack test cloud environment and the sequence of actions required for the deployment of both OpenStack itself and the Slurm virtual cluster environment for portable, scalable, reproducible scientific biomedical computing are presented.

The purpose of the paper is a description of the experience of test deployment of OpenStack to create a scalable computing environment for reproducible scientific computing using modern technological solutions, which can be applied to both cloud (OpenStack, AWS, Google) and cluster platforms (Slurm).

Results. The structure of the created test containerized (using Singularity technology) biomedical application which contains modern software and libraries and can be used in conventional and cloud virtual cluster environments is briefly described. The results of a comparative test of this application in the virtual cluster environment Slurm under the control of OpenStack and in the node of cluster SKIT-4.5 in the V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine are given. Information on solving the problem of finding the optimal in terms of saving resources scaling parameters for the developed application in two comparable cluster environments is given. Some features of the use of these cluster environments are clarified, in particular, a comparison of the dependence of the application speed on the number of parallel processes for two cluster environments is presented. Empirical data are presented in graphical form, which illustrate the nature of the load on the OpenStack server and the use of RAM on the number of parallel processes. Possibilities of portability between the specified cluster environments, scaling of calculations and maintenance of reproducibility of calculations for the offered test application are demonstrated. The advantages of using OpenStack technology for scientific biomedical calculations are pointed out.

Conclusions. The described example of test deployment and use of OpenStack gives an idea of the requirements for the necessary technical base to ensure the reproducibility of scientific biomedical calculations in cloud and cluster environments.

Keywords: cloud technologies, reproducible calculations, cluster platform.