

РОЗДІЛ IV. ІНФОРМАЦІЙНО-КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

УДК 004.272.2:004.75

DOI: 10.25140/2411-5363-2017-3(9)-88-97

Володимир Казимир, Ольга Пріла, Микола Крищенко

ЗАСТОСУВАННЯ ДИНАМІЧНИХ ВІРТУАЛЬНИХ ОБРАЗІВ У ГРІД-СЕРЕДОВИЩІ З ПІДТРИМКОЮ РЕПЛІКАЦІЇ

Актуальність теми дослідження. Нині грід-технології стрімко розвиваються в різних наукових галузях. Технологія віртуалізації дозволяє використовувати ліцензовані програмні забезпечення при обчисленні задач у грід-середовищі. Однак завантаження віртуального образу на віддалений ресурс значно впливає на час виконання грід-задачі. Також не вирішувалось питання налаштування вже сформованого образу віртуальної машини на віддаленому грід-ресурсі. Тому існує потреба в розробці оптимізованого розподіленого сховища для віртуальних образів у грід-середовищі.

Постановка проблеми. Сучасним проміжним програмним забезпеченням грід не вирішується задачі розподіленого збереження та налаштування динамічних віртуальних образів, на яких ґрунтується розгортання потрібного користувачеві оточення для виконання грід-задач.

Аналіз останніх досліджень і публікацій. Аналіз літературних джерел показав, що існуючі методи віртуалізації в грід-середовищі передбачають використовувати однотипні віртуальні образи для виконання грід-задач, не враховуючи оптимізації розподіленого сховища для образів віртуальних машин.

Виділення не досліджених частин загальної проблеми. У випадку пересилання образу віртуальної машини для подальшого розгортання на віддаленому ресурсі, час виконання задачі в грід-середовищі значно перевищує час при обчислюванні цієї ж задачі з попередньо встановленим віртуальним образом на грід-ресурсі. На сьогодні не вирішувалось питання оптимізації зберігання динамічних віртуальних образів у грід-середовищі.

Постановка завдання. Метою статті є розробка архітектури розподіленого сховища динамічних віртуальних образів у грід-середовищі з підтримкою механізму реплікації.

Виклад основного матеріалу. Проведено аналіз існуючих технологій використання віртуальних машин у грід-середовищі. Розробляється практична реалізація на базі існуючого фреймворку для розробки грід-додатків із використанням технології віртуалізації. Процес розгортання динамічного віртуального образу на віддалених обчислювальних ресурсах реалізується за допомогою програмної платформи Docker. Такий підхід до формування образів дає гнучкість в управлінні динамічними віртуальними образами і дозволяє з легкістю переносити вже налаштовані образи між обчислювальними ресурсами за допомогою GridFtp протоколу. В статті представлено механізм зберігання, доступу та реплікації динамічних віртуальних образів у грід-середовищі.

Висновки. Авторами розроблено алгоритм виконання грід-задачі з використанням динамічних образів віртуальної машини та архітектуру розподіленого сховища для зберігання динамічних віртуальних образів у грід-середовищі з підтримкою механізму реплікації.

Ключові слова: грід-середовище; віртуалізація; динамічний віртуальний образ; розподілене сховище; реплікація.
Рис.: 2. Табл.: 1. Бібл.: 29.

Постановка проблеми. Нині грід-технології стрімко розвиваються та застосовуються для високопродуктивних обчислень, а також значно поширені в наукових дослідженнях, медицині, економіці та інших галузях, де потрібно швидко, а головне дешевше обчислити великі масиви даних [1]. У грід кожному користувачу надається можливість використовувати певні прикладні програми (ПП) з відповідними налаштуваннями на віддалених ресурсах. Як правило, ці ресурси представлені кластерами, що працюють під управлінням ОС сімейства Linux.

Однак при вирішенні грід-задач часто виникає необхідність їх виконання під операційною системою Windows. Ця проблема традиційно вирішується за допомогою технології віртуалізації, яка дозволяє сформувану образ віртуальної машини (ВМ) з необхідними налаштуваннями. Але існує декілька невирішених задач щодо використання технології віртуалізації в грід-середовищі:

– завантаження віртуального образу (ВО) на віддалений ресурс грід-середовища займає багато часу (в середньому від однієї години і більше) [2], що значно збільшує час виконання грід-задач;

– налаштування раніше сформованого ВО під умови виконання конкретної задачі.

Сучасним проміжним програмним забезпеченням (ППЗ) грід не надається можливість автоматизованого налаштування ВО для виконання окремих задач. У той же

час ця проблема може бути якоюсь мірою вирішена шляхом реалізації в грід механізму реплікації, який з успіхом вже використовується у хмарних технологіях, таких як: Microsoft Azure [3], De Novo [4] та ін. Але реплікацію у хмарних технологіях застосовують тільки для резервного копіювання та синхронізації образів ВМ, не підтримуючи безпосередньо процес обчислень на розгорнутих ВО.

Аналіз останніх досліджень і публікацій. На сьогодні існують деякі рішення щодо використання віртуальних образів у грід-середовищі. Так, у статті [5] пропонується використання ВО, яке можна завантажити на ресурс по грід-мережі. Але цей ВО може бути використаний тільки для однотипних задач без можливості його налаштування під інші програми.

У роботі [6] розглядається варіант попереднього неавтоматичного розгортання ВО на віддаленому грід-ресурсі, і задача може бути запущена тільки на вже підготовленому ВО. Також у цьому прикладі на враховується можливість створення розподілених сховищ ВО, що може значно покращити часові характеристики при виконанні задач грід-середовищі.

У статті [7] вирішується задача розгортання ВМ на віддаленому грід-ресурсі, але не передбачається можливість налаштування сформованого ВО та його реплікації.

Виділення не вирішених раніше частин загальної проблеми. Авторами статті було оцінено ефективність технології побудови віртуального оточення для грід-застосувань. Результати експериментів на базі комплексу імітаційних моделей процесу виконання задачі в грід-середовищі з використання технології віртуалізації показали, що тимчасові витрати на розгортання ВО під час запуску задачі не перевищують 5 % у разі наявності заздалегідь визначеного ВО на грід-ресурсу. Однак відношення між часом виконання задачі в грід-середовищі без використання технології віртуалізації і з використанням, але без встановленого ВО на віддалений обчислювальний ресурс значно вище [2]. Але витрати на пересилку ВО можуть бути компенсовані за допомогою розробки розподіленого сховища для зберігання ВО в грід-середовищі.

Мета статті. У статті пропонується архітектура розподіленого сховища динамічних віртуальних образів у грід-середовищі з підтримкою механізму реплікації.

Аналіз існуючих технологій використання ВО в грід-середовищі. Під віртуалізацією розуміється абстракція обчислювальних ресурсів і надання користувачеві системи, яка «інкапсулює» (приховує в собі) власну реалізацію [8]. Ідея використання технології віртуалізації для гнучкого конфігурування потрібного користувачеві оточення при виконанні задач у грід-середовищі не є новою. Найвідомішими проектами з використання віртуалізації в грід-середовищах є RainBow [5], СКІФ-ГРІД [9] та Віртуальний «контейнер» [7].

Комплекс програмних засобів Rainbow («ARC in the Cloud») розроблений у межах проекту Український національний грід (УНГ) [10] для завдань інтерактивного аналізу медичних даних віртуальної організації «Медгрід» [11]. Спрямований на прискорення апаратного забезпечення віртуальних машин на робочих вузлах кластерів і орієнтований на використання ППЗ Nordugrid ARC [12]. Як обробник обчислювального елементу(СЕ, англ. Computing Element), а також сценарій керуючого циклом роботи віртуальної машини використовувався механізм RunTimeEnvironment (RTE) ПЗ Nordugrid ARC.

У таблиці представлено порівняльний аналіз даних прикладів використання віртуальних машин у грід-середовищі за певними критеріями.

Реалізація використовує готові елементи мережевої інфраструктури ОС Linux. Інтерактивний доступ до платформ на базі Windows здійснюється за допомогою протоколу віддаленого підключення до робочого столу RDP [13].

*Порівняльний аналіз існуючих технологій використання
віртуальних машин у грід-середовищі*

Критерії оцінки	Rainbow	СКІФ-ГРІД	Віртуальний «контейнер»
Передача ВО на віддалений ресурс	+	-	+
Зберігання ВО в розподіленому сховищі	+	-	+
Вибір ВО для виконання задачі	-	-	-
Наявність засобів реплікації ВО	-	-	-
Розширюваність	-	-	-

Примітки:

1. "-" означає відсутність реалізації цієї вимоги у розглянутому рішенні;
2. "+" означає наявність реалізації цієї вимоги.

Однак розглянуте рішення орієнтоване на запуск і виконання задач аналізу медичних даних, і передбачає використання однотипного ВО, яке можна завантажити на ресурс по мережі в грід-середовищі. Комплекс програмних засобів Rainbow не враховує можливість розширюваності ВО.

Проміжне програмне забезпечення СКІФ-ГРІД призначене для об'єднання розподілених високопродуктивних обчислювальних ресурсів в єдину грід-мережу, а також для організації доступу користувачів до ВО на віддалених грід-ресурсах.

На відміну від ППЗ, включеного в ЕМІ (англ. European Middleware Initiative) [14] стандарт, компоненти СКІФ-ГРІД реалізують можливість організації віртуальних сервісів.

Компоненти СКІФ-ГРІД:

- skifGrid-srv призначений для віртуалізації серверного обладнання і системного управління ВО, використовуючи реалізацію ППО Ganeti [15];
- skifgrid-crm - менеджер обчислювальних ресурсів (англ. Computing Element, CE) використовує ППЗ QosCosGrid [16];
- брокер CE (skifgrid-crb);
- клієнт консольного доступу до ресурсів (skifgrid-cli);
- моніторинг грід-мережі (skifgrid-mon).

ППЗ СКІФ-ГРІД є вільним програмним забезпеченням, яке дозволяється поширювати і вносити зміни відповідно до умов стандартної громадської ліцензії GNU (GPL) версії 3 [17].

Однак таке програмне рішення не враховує передачу ВО на віддалений грід-ресурс для виконання задачі. СКІФ-ГРІД також не враховує наявність сховища ВО на грід-вузлах.

У Науково-дослідному обчислювальному центрі МГУ ім. М. В. Ломоносова велися розробки з використання віртуальних машин у грід-технологіях при вирішенні задач в обчислювальній хімії. У цій концепції описано використання віртуальних машин (ВМ) у розподілених середовищах. Таким чином, користувачеві розподіленого середовища може бути надане повністю ізольоване віртуальне обчислювальне середовище (віртуальний «контейнер») [7]. За своїми властивостями віртуальний «контейнер» не поступається фізичному серверу. У цьому додатку використовується технологія віртуалізації, що у свою чергу дає незалежність від ОС і оточення. Також враховується повне відділення як конкретної служби або програми від зовнішнього середовища, так і зовнішнього середовища від неї.

Користувач отримує можливість створити образ віртуальної машини з попередньо встановленою ОС і повністю сконфігурованими додатками, націленими на вирішення

TECHNICAL SCIENCES AND TECHNOLOGIES

конкретної задачі. Цей образ потім передається на розподілений ресурс і розгортається там як грід-застосування, не вимагаючи налаштування даного вузла під конкретні задачі. Це істотно полегшує адаптацію ПП для роботи в грід-середовищі [7]. Додатковим плюсом є те, що такі технології загалом дозволяють запускати образи ВМ з операційними системами, відмінними від встановлених на ресурсах.

Однак цей метод розглянуто з урахуванням використання віртуальних машин для вирішення конкретних задач в обчислювальній хімії. Програмний додаток віртуальний «контейнер» вирішує задачу розгортання ВМ на віддаленому грід-ресурсі, але не враховує можливість розширюваності сформованого ВО та його реплікації.

Механізм реплікації та особливості його використання при віртуалізації.

Реплікація – це процес, під яким розуміється копіювання даних з одного ресурсу на інший (або на безліч інших) і навпаки [18]. При реплікації зміни, зроблені в одній копії об'єкта, можуть бути поширені між іншими копіями (репліками). Реплікація може бути синхронною або асинхронною.

При синхронній реплікації, якщо ця репліка оновлюється, всі інші репліки того ж фрагмента даних також повинні бути оновлені в тій самій транзакції. Але синхронна реплікація має недолік, бо вона створює додаткове навантаження при виконанні всіх транзакцій, в яких оновлюються будь-які репліки.

У разі асинхронній реплікації оновлення однієї репліки поширюється на інші через деякий час, а не в тій же транзакції. Таким чином, при асинхронній реплікації вводиться затримка, або час очікування, протягом якого окремі репліки можуть бути фактично неідентичними.

Реплікація широко використовується в системах управління базами даних (СУБД), де забезпечує такі можливості [19]:

- синхронізація даних. Дані розподілені по декількох ресурсів, кожен з яких підтримує актуальну копію інформації. Зміни на одному ресурсі поширюються між іншими ресурсами;
- інтеграція даних. Інформація, що збирається з декількох джерел, комбінується і надходить на множену інших ресурсів;
- консолідація даних. Набори даних на віддалених ресурсах, які відрізняються за контекстом, витягуються й об'єднуються на одному або декількох ресурсах.
- перенесення даних. У цьому випадку інформація переноситься зі старих ресурсів на нові.

Як вже вказувалось, реплікація знаходить широке використання у хмарних обчисленнях. Реалізація реплікації у хмарних технологіях забезпечує захист інформації від втрати у випадку виходу з ладу обладнання на кластері. У хмарі також надається можливість використовувати технологію віртуалізації з підтримкою механізму реплікації [20].

При віртуалізації реплікація надає такі можливості:

- періодично синхронізує копію образу ВМ з базовим, підтримуючи дані в актуальному стані;
- знімає питання падіння продуктивності обчислювальних ресурсів, оскільки репліка є копією віртуальної машини, яку можна запустити в будь-який момент часу;
- якщо базовий образ ВМ з якоїсь причини перестане працювати, надається можливість швидко перемкнутися на репліку і відновити потрібні ПП з мінімальним простоем;
- створення копій образу ВМ на локальний обчислювальний ресурс або на віддалений після аварійного відновлення.

Архітектура розподіленого сховища для зберігання динамічних віртуальних образів у грід-середовищі з підтримкою механізму реплікації. Архітектура, що пропонується, має за мету поєднання переваг застосування реплікації у СУБД та у розподіленому хмарному середовищі для підтримки виконання грід-задач.

Під ДВО розуміється образ віртуальної машини, який дає можливість розширювати необхідні користувачеві конфігурації оточення та встановлювати специфічні ПЗ.

Алгоритм виконання задачі з використанням ДВО в ґрід-середовищі представлений на рис. 1.

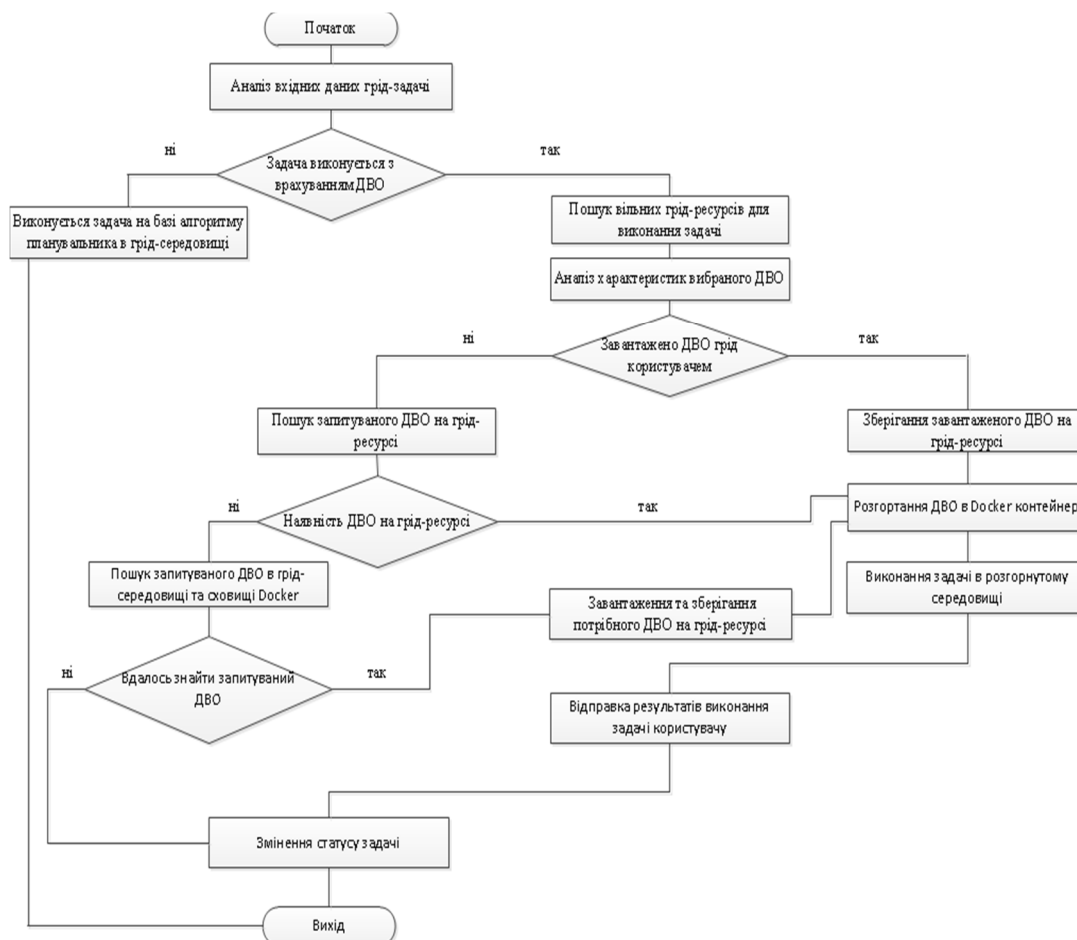


Рис. 1. Алгоритм виконання задачі з використанням ДВО в ґрід-середовищі

Процес розгортання ДВО на віддалених СЕ реалізується за допомогою програмної платформи Docker. Docker – програмне забезпечення для автоматизації розгортання і управління додатками в середовищі віртуалізації на рівні ОС [21].

Клієнтська сторона фреймворку дозволяє користувачам ґрід-порталу використовувати різні платформи, включаючи мобільні пристрої. При використанні технології віртуалізації для вирішення ґрід-задачі прикріплюється ДВО, якщо вибраний образ завантажується вперше, а також Dockerfile [21].

Dockerfile містить набір інструкцій з аргументами для розгортання ДВО. Після відправки задачі на виконання в ґрід-середовищі, виконуються сервіси Docker, які розгортають ДВО на основі інструкцій в Dockerfile. За допомогою інструкції TAG кожен ДВО ідентифікується унікальним ключем і заноситься в базу даних існуючого фреймворку [22].

Подібно віртуальній машині Docker запускає свої процеси у власній, заздалегідь налаштованій ОС. Але при цьому всі процеси Docker працюють на фізичному ґрід-сервері, ділячи всі процесори і всю доступну пам'ять з усіма іншими процесами. Підхід, використовуваний Docker знаходиться посередині між запуском всього на фізичному сервері і повної віртуалізації, пропонованої віртуальними машинами. Цей підхід називається контейнеризацією.

Під час виконання ґрід-задачі, на основі вибраного ДВО, у docker контейнері створюється віддалена ОС з потрібними для користувачеві налаштуваннями та застосуваннями. Самі ж результати виконання задачі будуть автоматично завантажено на

TECHNICAL SCIENCES AND TECHNOLOGIES

клієнтську частину користувача. Розширенням можливостей у вже встановленому ДВО на віддаленому грід-ресурсу займається розробник грід-сервісів. Розробник має RDP доступ до віддаленої ОС для створення нових або редагування вже існуючих грід-сервісів. Під грід-сервісами розуміється сценарій автоматизованого виконання задачі всередині ОС Docker контейнера.

На рис. 2 представлена архітектура розподіленого сховища для зберігання динамічних віртуальних образів в грід-середовищі з підтримкою механізму реплікації.

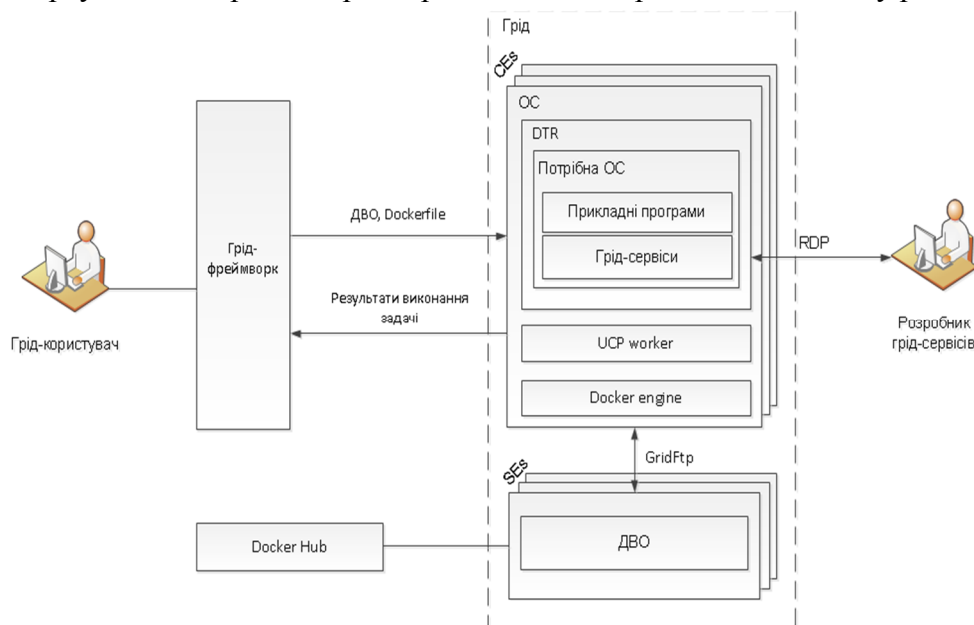


Рис. 2. Архітектура розподіленого сховища для зберігання динамічних віртуальних образів у грід-середовищі з підтримкою механізму реплікації

Docker Trusted Registry (DTR) – це контейнерні додатки, що запускаються на кластері Docker Universal Control Plane (UCP). За замовчуванням DTR зберігає ДВО на файловій системі SE (англ. Storage Element) грід-вузла. DTR також дає можливість використання централізованого резервного копіювання (реплікації) ДВО. DTR підтримує наступні технології для зберігання резервної копії ДВО: NFS [23], Amazon S3 [24], Cleversafe [25], Google Cloud Storage [26], OpenStack Swift [27] та Microsoft Azure.

Платформа Docker дає можливість використовувати власне сховище зареєстрованих ДВО. Таке сховище називається Docker hub.

ДВО складається з декількох образів файлової системи. Наприклад, перший образ може містити тільки базову установку ОС, другий додає до неї набір стандартних демонів, третій – утиліти адміністрування і так далі. Docker монтує всі шари в режимі «тільки читання», але так, щоб розробник грід-сервісів мав можливість змінювати вміст ДВО.

За замовчуванням після завершення DTR (яке відбувається після завершення останнього працюючого в ньому процесу) останній шар стирається і всі зміни пропадають. Однак, використовуючи команду `docker commit`, розробник грід-сервісів може «зафіксувати» зміни, створивши новий ДВО на основі вже існуючих образів.

Такий підхід до формування образів дає гнучкість в управлінні ДВО і дозволяє з легкістю переносити вже сконфігуровані ДВО між машинами за допомогою GridFtp [28] (образ можна викласти на Docker Hub і потім розгорнути на іншому грід-вузлі). Також Docker дозволяє економити дисковий простір, у випадку розгортання на грід-вузлі декількох ДВО. Кожен ДВО буде спочатку заснований на одному базовому образі. Отже, ДВО будуть посилатися на цей базовий образ і не будуть дублювати його вміст.

Запропонована технологія [29] передбачає високорівневий інтерфейс для грід-користувачів. Що дасть можливість задавати необхідні налаштування для віртуального середовища, в якому будуть виконуватись грід-задачі.

Практична реалізація запропонованої архітектури розподіленого сховища з підтримкою механізму реплікації здійснена на базі існуючого фреймворку для розробки грід-застосунків [22] з використанням технології побудови віртуального оточення [29].

Висновки. Проведено порівняльний аналіз технологій використання віртуалізації в грід-середовищах. Розроблено алгоритм виконання задачі з використанням динамічних віртуальних образів у грід-середовищі.

У статті запропоновано архітектуру розподіленого сховища для зберігання динамічних віртуальних образів у грід-середовищі з використанням Docker інструментів. Запропонована архітектура дозволяє реалізувати механізм реплікації динамічних віртуальних образів та використовувати їх як базу для інших образів, що економить час на завантаження та пам'ять на грід-ресурсі. Такий підхід значно зменшує час виконання задачі в грід-середовищі з використанням технології віртуалізації.

Подальші дослідження будуть спрямовані на розробку алгоритму планування виконання грід-задачі з урахуванням використання засобів віртуалізації, а також проведення експериментів в реальному грід-середовищі для визначення ефективності використання технології побудови віртуального оточення для грід-застосунків.

Список використаних джерел

1. *Grid* як четвертий етап розвитку інформатизації [Електронний ресурс]. – Режим доступу : <http://grid.kpi.ua/index.php/uk/what-is-grid/2-grid-yak-chetvertii-etap-rozvitku-nformatizac-.html>.
2. *Building of the virtual environment for grid applications* / V. Kazymyr, D. Melnychenko, O. Prila, M. Kryshchenko // *Information Models and Analyses*. – 2016. – Vol. 5, № 1. – С. 37–48.
3. *Microsoft Azure* [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/azure>.
4. *De Novo* [Електронний ресурс]. – Режим доступу : <https://www.de-novo.biz/uk/oblachnye-servisy>.
5. *Сальников А. А.* «Платформа как сервис» в грід для інтерактивного аналізу медичинських даних / А. А. Сальников, В. В. Вишневіський, А. Ф. Борецький // *Математичні машини і системи*. – 2015. – № 1. – С. 53–64. – Режим доступу : http://www.immsp.kiev.ua/publications/articles/2015/2015_1/01_2015_Salnikov.pdf.
6. *Московський А. А.* Віртуальні інструменти для грід-випрацювань / А. А. Московський, А. Ю. Первін, В. Walker // *Труди Другої Міжнародної наукової конференції «Суперкомп'ютерні системи і їх застосування (SSA'2008)2*. – 2008. – С. 185–188.
7. *Virtual computing environments: the use of polygons on the grid* / V. Volohov, D. Varlamov, N. Surkov [et al.] // *Herald of South Ural State University. Series: Mathematical modeling and programming*. – 2009. – № 17. – С. 24–35.
8. *Богданов А. В.* Віртуалізація: нові можливості відомої технології / А. В. Богданов, Е. Н. Станкова, В. В. Марєєв // *Всеросійський конкурсний відбір оглядово-аналітичних статей по пріоритетному напрямку «Інформаційно-телекомунікаційні системи»*. – 2008. – 31 с. – Режим доступу : <http://window.edu.ru/resource/802/58802/files/68359e2-st15.pdf>.
9. *SkifGrid* wiki [Електронний ресурс]. – Режим доступу : <http://grid.basnet.by/projects/skifgrid/wiki>.
10. *Український національний грід* [Електронний ресурс]. – Режим доступу : http://ung.in.ua/upload/user_files/BCC_Policy/ung_fin.pdf.
11. *Vishnevsky V.* Web-services of Medgrid project / V. Vishnevsky, M. Volzheva, O. Prila // *Ukrainian Journal of Telemedicine and medical telematics*. – 2012. – Vol. 10, № 2. – С. 4–5.
12. *NorduGrid ARC* [Електронний ресурс]. – Режим доступу : <http://www.nordugrid.org/manuals.html>.
13. *Remote Desktop Protocol* [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/Remote_Desktop_Protocol.
14. *European Middleware Initiative* [Електронний ресурс]. – Режим доступу : https://www.dcache.org/manuals/EMI_FACT-SHEET-1_4.pdf.
15. *Ganeti* [Електронний ресурс]: – Режим доступу: <http://www.ganeti.org>.

TECHNICAL SCIENCES AND TECHNOLOGIES

16. *QCG - Quality in Cloud and Grid* [Електронний ресурс]. – Режим доступу : <http://www.qoscosgrid.org/trac/qcg>.
17. *Загальна громадська ліцензія GNU* [Електронний ресурс]. – Режим доступу : <http://www.gnu.org/licenses/gpl-3.0.html>.
18. *Репликація* (вычислительная техника) [Електронний ресурс]. – Режим доступу : [https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_\(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0\)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)).
19. *Репликация баз данных* [Електронний ресурс]. – Режим доступу : https://www.sybase.ru/system/files/pdf/sybase_rs_db_replication_wp_ru_2.pdf.
20. *Сравнение решений для backup виртуальных машин от VMware, Veeam, Acronis и Symantec* [Електронний ресурс]: – Режим доступу: <http://www.issystems.kz/article/4/sravnenie-reshenii-dlya-backup-virtualnykh-mashin-ot-vmware-veeam-acronis-i-symantec>.
21. *Docker Overview* [Електронний ресурс]. – Режим доступу : <https://docs.docker.com/engine/understanding-docker>.
22. *Prila O. The framework for high level grid applications development / O. Prila // Problems of programming. – 2014. – № 1. – С. 31–39.*
23. *Network File System (NFS)* [Електронний ресурс]. – Режим доступу : <http://searchenterprisedesktop.techtarget.com/definition/Network-File-System>.
24. *Amazon S3* [Електронний ресурс]. – Режим доступу : https://aws.amazon.com/ru/s3/?sc_channel=PS&sc_campaign=acquisition_UA&sc_publisher=google&sc_medium=s3_b&sc_content=s3_e&sc_detail=amazon%20s3&sc_category=s3&sc_segment=192085788688&sc_matchtype=e&sc_country=UA&sc_kwid=AL!4422!3!192085788688!e!!g!!amazon%20s3&ef_id=WcAQHgAABFtWoE1D:20170925083959:s.
25. *IBM Cloud Object Storage* [Електронний ресурс]. – Режим доступу : <https://www.ibm.com/cloud-computing/products/storage/object-storage>.
26. *Google Cloud Storage* [Електронний ресурс]. – Режим доступу : <https://cloud.google.com/storage>.
27. *OpenStack Object Storage ("Swift")* [Електронний ресурс]. – Режим доступу : <https://wiki.openstack.org/wiki/Swift>.
28. *GT 6.0 GridFTP* [Електронний ресурс]. – Режим доступу : <http://toolkit.globus.org/toolkit/docs/latest-stable/gridftp>.
29. *Казимир В. В. Архітектура побудови віртуального оточення для ґрід-застосувань / Казимир В. В., Пріла О. А., Крищенко М. С. // Математичні машини і системи. – 2017. – № 3. – С. 65–73.*

References

1. *Grid iak chetvertii etap rozvitku informatizats [Grid as the fourth stage of development of Informatization]*. Retrieved <http://grid.kpi.ua/index.php/uk/what-is-grid/2-grid-yak-chetvertii-etap-rozvitku-nformatizac-.html>.
2. Kazymyr V., Melnychenko D., Prila O., Kryshchenko M. (2016). Building of the virtual environment for grid applications. *Information Models and Analyses*, vol. 5, no. 1, pp. 37–48.
3. *Microsoft Azure*. Retrieved from <https://docs.microsoft.com/en-us/azure>.
4. *De Novo*. Retrieved from <https://www.de-novo.biz/uk/oblacloudnye-servisy>.
5. Salnikov, A., Vishnevskii, V., Boretskii, A. (2015). Platforma kak servis v grid dlia interaktivnogo analiza meditsinskikh dannykh [Platform as a service" in the grid for interactive analysis of medical data]. *Mathematical machines and systems*, no. 1, pp. 53–64. Retrieved from http://www.immsp.kiev.ua/publications/articles/2015/2015_1/01_2015_Salnikov.pdf.
6. Moskovskii A. Pervin A. Walker B. (2008) Virtualnye instrumenty dlia grid-vychislenii [Virtual tools for grid computing]. Proceedings of the *Second International Scientific Conference "Supercomputer systems and their application (SSA'2008)"*, pp. 185–188.
7. Volohov, V., Varlamov, D., Surkov, N. (2009). Virtualnye vychislitelnye sredy ispolzovaniia na grid poligonakh [Virtual computing environments: utilization on the grid polygons]. *Herald of South Ural State University Series: Mathematical modeling and programming*, no. 17, pp. 24–35 (in Russian).
8. Bogdanov, A., Stankova, E., Mareev, V. (2008). Virtualizatsiia novye vozmozhnosti izvestnoi tekhnologii [Virtualization: new possibilities of the known technology]. *All-Russian competitive*

selection of review-analytical articles on the priority direction "Information and telecommunication systems", p. 31. Retrieved from <http://window.edu.ru/resource/802/58802/files/68359e2-st15.pdf>.

9. *SkifGrid wiki*. Retrieved from <http://grid.basnet.by/projects/skifgrid/wiki>.

10. *Ukrainskii natsionalnii grid [Ukrainian national grid]*. Retrieved from http://ung.in.ua/upload/user_files/BCC_Policy/ung_fin.pdf.

11. Vishnevsky, V., Volzheva, M., Prila, O. (2012). Web-services of Medgrid project [Web-services of Medgrid project]. *Ukrainian Journal of Telemedicine and medical telematics*, vol. 10, no. 2, pp. 4–5.

12. *NorduGrid ARC*. Retrieved from <http://www.nordugrid.org/manuals.html>.

13. *Remote Desktop Protocol*. Retrieved from https://uk.wikipedia.org/wiki/Remote_Desktop_Protocol.

14. *European Middleware Initiative*. Retrieved from https://www.dcache.org/manuals/EMI_FACT-SHEET-1_4.pdf.

15. *Ganeti*. Retrieved from <http://www.ganeti.org/>.

16. *QCG – Quality in Cloud and Grid [Quality in Cloud and Grid]*. Retrieved from <http://www.qoscosgrid.org/trac/qcg>.

17. *Zagalna gromadska licenziia GNU [General public license GNU]*. Retrieved from <http://www.gnu.org/licenses/gpl-3.0.html>.

18. *Replikatsiia (vychislitelnaia tekhnika) [Replication (computer technology)]*. Retrieved from [https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_\(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0\)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)).

19. *Replikatsiia baz dannykh [Database replication]*. Retrieved from https://www.sybase.ru/system/files/pdf/sybase_rs_db_replication_wp_ru_2.pdf.

20. *Sravnienie reshenii dlia backup virtualnykh mashin ot VMware, Veeam, Acronis i Symantec [Comparison of solutions for backup virtual machines from VMware, Veeam, Acronis and Symantec]*. Retrieved from <http://www.issystems.kz/article/4/sravnienie-reshenii-dlya-backup-virtualnykh-mashin-ot-vmware-veeam-acronis-i-symantec>.

21. *Docker Overview*. Retrieved from <https://docs.docker.com/engine/understanding-docker>.

22. Prila, O. (2014). The framework for high level grid applications development. *Problems of programming*, no. 1, pp. 31–39.

23. *Network File System (NFS)*. Retrieved from <http://searchenterprisedesktop.techtarget.com/definition/Network-File-System>.

24. *Amazon S3*. Retrieved from https://aws.amazon.com/ru/s3/?sc_channel=PS&sc_campaign=acquisition_UA&sc_publisher=google&sc_medium=s3_b&sc_content=s3_e&sc_detail=amazon%20s3&sc_category=s3&sc_segment=192085788688&sc_matchtype=e&sc_country=UA&sc_kwcid=AL!442213!192085788688!e!g!amazon%20s3&ef_id=WcAQHgAABFtWoE1D:20170925083959:s.

25. *IBM Cloud Object Storage*. Retrieved from <https://www.ibm.com/cloud-computing/products/storage/object-storage>.

26. *Google Cloud Storage*. Retrieved from <https://cloud.google.com/storage>.

27. *OpenStack Object Storage ("Swift")*. Retrieved from <https://wiki.openstack.org/wiki/Swift>.

28. *GT 6.0 GridFTP*. Retrieved from <http://toolkit.globus.org/toolkit/docs/latest-stable/gridftp>.

29. Kazymyr, V., Prila, O., Kryshchenko, M. (2017). Arkhitektura pobudovi virtualnogo otochennia dlia grid-zastosuvan [The architecture of virtual environment for grid applications]. *Mathematical machines and systems*, no. 3, pp. 65–73.

UDC 004.272.2:004.75

Volodymyr Kazymyr, Olga Prila, Mykola Kryshchenko

THE USE OF DYNAMIC VIRTUAL IMAGES IN A GRID ENVIRONMENT WITH REPLICATION SUPPORT

Urgency of the research. Currently grid technologies are rapidly evolving in different scientific fields. Virtualization technology allows you to use the licensed software for computing tasks in a grid environment. However, the download of the virtual image on the remote resource significantly increases the execution time of grid tasks. Also, the issue of configuration the generated image of a virtual machine on a remote grid resource was not solved. There is therefore a need to develop optimized distributed storage for virtual images in a grid environment.

Target setting. Modern middleware grid does not solve the problem of distributed storage and configuration of dynamic virtual images, based on the deployment of the necessary environment user to perform grid tasks.

Actual scientific researches and issues analysis. The analysis of literary sources showed that existing methods of virtualization, grid environments involve the use of similar virtual images to perform a grid task, without considering the optimization of the distributed storage of virtual machine images.

Uninvestigated parts of general matters defining. Transfer the virtual image to a grid resource is time-consuming. This time considerably exceeds the time of task execution on grid resource, which has a pre-installed virtual image. To date, the issue of optimizing the storage of dynamic virtual images in the grid environment has not been resolved.

The research objective. The purpose of this paper is to develop architecture of distributed storage for dynamic virtual images in a grid environment with support for replication.

The statement of basic materials. The technology implementation is done on the basis of the existing framework for grid applications development. The mechanism for realizing the deployment of a virtual image on remote computing resources is implemented on software Docker platform. This approach to the formation of images gives the flexibility to manage dynamic virtual images and allows you to easily transfer already configured images between computing resources by using GridFtp Protocol. The article presents a mechanism for storing, accessing and replicating dynamic virtual images in a grid environment.

Conclusions. The authors research the existing technologies for building a virtual environment for computing tasks on the remote resource of the distributed environment. The architecture of distributed storage for storing virtual images in a grid environment with support for replication was designed.

Key words: grid environment; virtualization; virtual image; distributed storage; replication.

Fig.: 2. Tabl.: 1. Bibl.: 29.

УДК 004.272.2:004.75

Владимир Казимир, Ольга Преляя, Николай Крищенко

ПРИМЕНЕНИЕ ДИНАМИЧЕСКИХ ВИРТУАЛЬНЫХ ОБРАЗОВ В ГРИД-СРЕДЕ С ПОДДЕРЖКОЙ РЕПЛИКАЦИИ

На данный момент грид-технологии стремительно развиваются в различных научных областях. Технология виртуализации позволяет использовать лицензированные программные обеспечения при вычислении задач в грид-среде. Однако загрузка виртуального образа на удаленный ресурс значительно влияет на время выполнения грид-задачи. Современным промежуточным программным обеспечением грид не решается задачи распределенного хранения и настройки динамических виртуальных образов, на которых основывается развертывания требуемого окружения для выполнения грид-задач. Процесс развертывания динамического виртуального образа на удаленных вычислительных ресурсах реализуется с помощью программной платформы Docker. В статье представлен механизм хранения, доступа и репликации динамических виртуальных образов в грид-среде. Авторами разработано алгоритм выполнения грид-задачи с использованием динамических образов виртуальных машин и архитектуру распределенного хранилища для хранения динамических виртуальных образов в грид-среде с поддержкой механизма репликации.

Ключевые слова: грид-среда; виртуализация; динамический виртуальный образ; распределенное хранилище; репликация.

Рис.: 2. Табл.: 1. Библ.: 29.

Казимир Володимир Вікторович – доктор технічних наук, професор, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Казимир Владимир Викторович – доктор технических наук, профессор, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Kazymyr Volodymyr – Doctor of Technical Sciences, Professor, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: vvkazymyr@gmail.com

ORCID: 0000-0001-8163-1119

Пріла Ольга Анатоліївна – кандидат технічних наук, доцент кафедри інформаційних та комп'ютерних систем, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Преляя Ольга Анатольевна – кандидат технических наук, доцент кафедры информационных та компьютерных систем, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Prila Olga – PhD in Technical Sciences, Associate Professor of Department of Information and Computer Systems, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: olga.prila1986@gmail.com

ORCID: 0000-0002-0884-6516

Крищенко Микола Сергійович – асистент кафедри інформаційних та комп'ютерних систем, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Крищенко Николай Сергеевич – ассистент кафедры информационных та компьютерных систем, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Mykola Kryshchenko – assistant of lecturer of Department of Information and Computer Systems, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: mykola3451@gmail.com

ORCID: 0000-0003-4783-9260