УДК 004.02

ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ВІРТУАЛЬНИХ МАШИН ДЛЯ РОЗРОБЛЕННЯ СУЧАСНИХ ВЕБ-РЕСУРСІВ ТА СЕРВІСІВ

Алексієв В.О.

Харківський національний економічний університет імені Семена Кузнеця, Харків

Сучасні веб-рішення можна поділити на два типи: веб-ресурси, які характеризуються наявністю інформаційної складової та веб-сервіси, що призначені для надання сервісних послуг й переважно реалізовані за допомогою технологій прогресивних веб-додатків (Progressive Web App, PWA). PWA – фактично є програмним застосунком, реалізованим на JavaScript або ін., який працює у браузері. Розробка відповідних систем потребує етапу прототипування продукту та тестування його властивостей з подальшою доставкою користувачу, як ресурсу або сервісу. Для рішення цього завдання найбільш раціональним є застосування технологій віртуальних машин [1].

Зараз технології віртуалізації набули значної популярності у рішенні будь-яких завдань у IT-галузі. Слід розділяти відповідні технології на два типи – віртуалізація рівня робочого стола (Desktop Virtualization) та серверна віртуалізація (Server Virtualization). Віртуалізація рівня робочого стола спрямована на створення віртуальної машини як копії операційної системи та її програмного оточення, що працюють на фізичному обладнанні.

Наприклад, проект QEMU (https://www.qemu.org/). Це відкрита система, з повною емуляцією обладнання. У неї кожна програмна команда гостьової операційної системи інтерпретується до команди основної системи, на якій працює система віртуалізації. Проект QEMU – це гарне рішення, його наробки також застосовуються у інших проектах віртуалізації для побудови емуляторів апаратного устаткування. Звичайно, це не найшвидший варіант віртуалізації, однак, він дозволяє запускати у якості віртуальної машини рішення, що принципово відрізняються від апаратної платформи основної хост-системи.

Визначені системи підходять для рішення завдань розробки веб-інтерфейсу апаратних маршрутизаторів, наприклад на базі RISC-процесорів тощо.

Інший підхід – це повна віртуалізація, де гостьові системи повинні бути розраховані на ту ж архітектуру апаратного устаткування (тип процесору), що й основна система. Оскільки у цих рішеннях відсутня переробка команд процесору гостьової операційної системи до команд основної системи, то швидкість роботи цього рішення, у порівнянні з емуляцією обладнання, є значною. Прикладом реалізації таких систем є VirtualBox (https://www.virtualbox.org/), VMware Workstation Pro й Workstation Player (https://www.vmware.com/products/workstation-player.html) та ін. Ці системи спрямовані на надання користувачу робочого простору середовища операційної системи. Вони зарекомендували себе у якості тестової платформи налагодження операційних систем та тестування веб-застосунків.

Наприклад, для VirtualBox можна залучити наробки системи Vagrant (https://www.vagrantup.com/), яка надає можливості автоматизувати процеси створення, запуску та повторного застосування віртуальних машин. Ці рішення можна вважати оптимальними для виконання тестування на рівні завдань системного адміністратора або розробника.

Фактично віртуальна машина є операційною системою, яка запущена у захищеному оточенні, а середовище віртуалізації виконує роль ізоляції гостьових систем та поділ ресурсів основної системи (дисковий простір, пам'ять та процесорний час). Також, середовище віртуалізації звуть гіпервізором, або програмою, яка управляє виконанням гостьових віртуальних машин.

Якщо у віртуалізації рівня робочого стола головним є виконання завдань користувача та швидкодія відбиття графічного оточення операційної системі, то засоби серверної віртуалізації розраховані на найменше споживання ресурсів самою системою віртуалізації та швидкодію роботи гостьових систем, доступ до яких, зазвичай, буде за мережевими протоколами доступу.

Тому на стороні серверу буде виконуватись інша система віртуалізації. Це протиріччя здебільше не є актуальним у разі застосування систем Hyper-V від Microsoft, технологій віртуалізації компанії VMware або гіпервізору Xen. Однак, перераховані системи більш притаманні завданням серверної віртуалізації, а на рівні робочого стола дуже часто застосовується система VirtualBox.

У разі гетерогенного середовища віртуалізації актуальним стає застосування систем автоматизації розгортання операційних систем. Наприклад, рішення Chef (https://www.chef.io/), Puppet (https://puppet.com/), Ansible (https://www.ansible.com/) та ін. дозволяють розробити скрипти розгортання необхідного оточення. Що дозволяє повторювати конфігурації, як: операційна система Ubuntu Server, її компоненти, база даних MySQL, веб-сервер Apache та його застосунки.

З точки зору уніфікації застосування різних систем віртуалізації доречним є визначення формату файлів віртуальної машини. Найбільш розповсюдженими є VMDK, який застосовується переважно у рішеннях VMware, формат VHD компанії Microsoft. Поруч з цим, наприклад, у системі VirtualBox застосовується формат VDI. Звичайно подолати складності застосування різних рішень віртуалізації можна завдяки конверторам форматів файлів віртуальних машин або застосуванню відкритого формату OVF (Open Virtualization Format), однак, він також підтримується не всіма сучасними системами. Таким чином, найбільш привабливим шляхом уніфікації процесів розгортання операційних систем є застосування спеціалізованих рішень автоматизації, які зазначені раніше.

Для рішення завдань на стороні сервера сьогодні застосують найрізноманітніші системи віртуалізації на основі гіпервізора: Hyper-V (https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview), Xen (https://xenproject.org/) [2], KVM (https:// www.redhat.com/ en/ topics/ virtualization/ what-is-KVM), рішення VMware (https://www.vmware.com/ solutions/ virtualization.html) та ін. Фактично такі

системи підтримують так-звану паравіртуалізацію, яка передбачає, що гостьова операційна система є модифікованою та знає про свою роботу у середовищі віртуалізації. Це надає безумовних переваг щодо швидкодії відповідних рішень.

Поруч з підтримкою апаратних технологій віртуалізації рівня процесору: Intel-VT та AMD-V, сучасні системи віртуалізації досягають рівня продуктивності порівняно з невіртуалізованою системою. Віртуалізовані системи дозволяють реалізувати парадігму: один сервер – один сервіс, а також більш ефективно за невіртуалізовані системи споживати ресурси апаратного устаткування. Наприклад, у багатьох системах реалізована можливість «живої міграції (live migration)», яка дозволяє перенести віртуальну машину на інший хост практично без втрати з'єднання [3]. Тому, серверна віртуалізація зараз є стандартом де-факто при побудові сучасних IT-рішень.

Впровадження новітніх веб-технологій у транспортних та промислових застосуваннях потребують більше ресурсів чим один сервер віртуалізації. Тому, для рішення завдань рівня побудови транспортного порталу доречно застосовувати кластери віртуальних машин, що будуються на залученні десятків фізичних серверів. Звичайно системи віртуалізації є безкоштовними на рівні одного серверу, у випадку координації управління багатьма системами застосовують комерційні рішення систем віртуалізації. Слід відзначити, що рішення Proxmox VE є відкритою системою та має варіант комерційної підписки, у разі потреби застосовувати перепідготовлені сертифіковані образи для створення віртуальних машин [4]. Proxmox VE дозволяє управляти кластерами віртуальних машин, що управляються системами віртуалізації KVM та LXC.

LXC (Linux Containers) – це віртуалізація рівня операційної системи. До таких систем відносять й технологію Docker (https://www.docker.com/). Вона відрізняється від традиційної віртуалізації на базі гіпервізора тим, що в цієї системі віртуальна машина є фактично ізольованою копією робочого простору користувача.

Контейнерна віртуалізація виконує ізоляцію не самої операційної системи, а безпосередньо застосунків, що пакуються у захищений контейнер. Такий підхід дає вищій рівень швидкодії віртуальної машини. Також засоби Docker надають автоматизації для створення та управління контейнерами, як у застосунках, що вирішують завдання серверу, так й для Desktop-рішень. Це вирішує визначену проблему гетерогенного простору застосування технологій віртуалізації (слід відзначити, що Docker підтримується як у Linux, так й Windows рішеннях). Однак, засоби Docker не завжди ефективні при рішенні завдань розгортання складних рішень для певної операційної системи, там де треба перевірити та виконати тести з конфігурування та налагодження компонентів встановленого застосунку тощо.

Таким чином, у багатьох рішеннях сучасні технології віртуалізації з різною архітектурою будуть застосовані при рішенні завдань розробки складних веб-орієнтованих систем. У більшості випадків залучення засобів керування конфігурацією операційних систем та програм вирішують проблеми сумісності обраних систем віртуалізації.

Список використаних джерел

[1] В. О. Алексієв, навч.-метод. посіб. Застосування GRID-технології у транспортному ВНЗ, Харків : ХНАДУ, 208 с. 2008.

[2] SUSE Linux Enterprise Server 15. Virtualization Guide, 2022 [Он-лайн]. Доступно: https://documentation.suse.com/sles/15-GA/html/SLES-all/book-virt.html.

[3] Red Hat Virtualization 4.4 Virtual Machine Management Guide. Red Hat Virtualization Documentation Team, 2022 [Он-лайн]. Доступно: https://access.redhat.com/documentation/ en-us/ red_hat_virtualization/ 4.4/ html/ virtual_machine_management_guide.

[4] Proxmox VE Administration Guide. Release 7.2, 535 p., 2022 [Он-лайн]. Доступно: https://www.proxmox.com/en/downloads/item/proxmox-ve-admin-guide-for-7-x.