

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

Магістр

на тему: **Методи та інструментальні засоби управління процесом міграції віртуальних машин в обчислювальній хмарі**

Виконав: студент 6 курсу, групи «ІСМ-6»
спеціальності

123 «Комп'ютерна інженерія»

Керівник

Нормоконтроль

Рецензент

Северин Д.А.

Баран І.О.

Тим С.В.

Гашин Н.Б.

Міністерство освіти і науки України
 Тернопільський національний технічний університет імені Івана Пулюя
(колишній національний технічний університет імені Івана Пулюя)

Факультет ФІІ
 Кафедра Комп'ютерних систем та мереж
 Освітньо-кваліфікаційний рівень Магістр
 Напрямок підготовки _____
 Спеціальність 123 Комп'ютерна інженерія
(код спеціальності)

ЗАТВЕРДЖУЮ

Завідувач кафедри Олена Василівна Т.М.
 « 30 » _____ 2019 р.

Імені Івана Пулюя

**ЗАВДАННЯ
 НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ**

Степану Дмитру Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Методи та інструментарій роботи з управління процесом міграції віртуальних машин в обчислювальній мережі

Керівник проекту (роботи) К.Т.Н. Олена Варан В.А.
(прізвище, ім'я, по батькові, науковий ступінь, очолює зміст)

Затверджені наказом по університету від « 27 » 09 2019 року № 417-019

2. Термін подання студентом проекту (роботи) 27.12.2019

3. Вихідні дані до проекту (роботи) Методи та інструментарій роботи з управління процесом міграції ВМ у хмарі

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

методи та інструментарій роботи з управління процесом міграції віртуальних машин в обчислювальній мережі. Аналіз програмної області. Особливості міграції віртуальних машин в обчислювальній мережі. Проектування системи управління процесом міграції ВМ у хмарі. Аналіз програмної області. Особливості міграції віртуальних машин в обчислювальній мережі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Методи та інструментарій роботи з управління процесом міграції віртуальних машин в обчислювальній мережі. Аналіз програмної області. Особливості міграції віртуальних машин в обчислювальній мережі. Проектування системи управління процесом міграції ВМ у хмарі. Аналіз програмної області. Особливості міграції віртуальних машин в обчислювальній мережі.

6. Консультанти розділів проєкту (роботи)


Розділ	Прізвище, ініціал та посада консультанта	Підпис, дата виконання завдання	Підпис, дата виконання завдання
Експерт охорони праці	Васюк О.В., доц Суховська Г.І.	[Підпис]	[Підпис]
Безпека в НС	Стручок В.С., ст. викл. карот	[Підпис]	[Підпис]
Об'єкти екон. ефект	Овчарук Н.Б.	[Підпис]	[Підпис]

7. Дата видачі завдання 30.09.2019

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Термін виконання етапів проєкту (роботи)	Підпис
1.	Отримання завдання	20.09.2019	
2.	Вибір тематичної області	20.10.19	
3.	Аналіз предметної області	28.10.19	
4.	Бюджетування міграції інтелектуальних машин в обчислювальній техніці	4.11.19	
5.	Практичне реалізація програмного модулю ХЕН М:ОР для управління професійною міграцією інтелектуальних машин в обчислювальній техніці	9.11.19	
6.	Обчислювальна економіка міграції в обчислювальній техніці	16.11.19	
7.	Охорона праці при роботі з електроенергією	27.11.19	
8.	Економіка	19.12.19	
9.	Підсумок роботи з виконання роботи	28.11.19	
10.	Захист дипломної роботи	22.12.19	

Студент  Сєверин Дмитро Іванович
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)  Баран І.О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Методи та інструментальні засоби управління процесом міграції віртуальних машин в обчислювальній хмарі // Дипломна робота за освітнім рівнем «магістр» // Северин Дмитро Анатолійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІм-62 // Тернопіль, 2019 // с. – , рис. – 24 , табл. – 6 , аркушів А1 – 10 , бібліогр. – 18.

Ключові слова: ВІРТУАЛІЗАЦІЯ, ГІПЕРВІЗОР, МІГРАЦІЯ, ПАРАВІРТУАЛІЗАЦІЯ, РЕСУРСИ, ХМАРА

Дипломна робота присвячена дослідженню сучасних технологій віртуалізації і засобів управління процесом міграції віртуальних машин в обчислювальній хмарі. Розроблений програмний засіб являє собою набір системних служб та призначених для користувача інтерфейсів, виконуються на реальних і віртуальних серверах обчислювального комплексу, який надає набір обчислювальних ресурсів, тобто хмари. Ці служби здійснюють збір інформації про використання різних системних ресурсів і відповідної завантаженості серверів і формують на основі цих даних статистику використання ресурсів, а також дозволяють здійснювати швидке розгортання віртуальних машин, автоматично створюваних на основі заданого шаблону, на реальних серверах і централізовано керувати міграцією наявних віртуальних машин на основі гіпервізора Xen між реальними серверами.

У першому розділі описано поняття віртуалізації, історія її розвитку, проведено огляд технологій віртуалізації. Основну увагу приділено аналізу процесу віртуалізації на основі VM. Наведено основні переваги та недоліки віртуалізації.

У другому розділі описано поняття гіпервізора та досліджені його типи. Проаналізовано основні функції та можливості найбільш вживаних гіпервізорів (Xen, VMware ESX Server, Hyper-V, KVM). Досліджено особливості міграції VM в обчислювальній хмарі.

У третьому розділі описано функціональні можливості та архітектура розробленого програмного засобу Xen Migr. Наведена програмна архітектура розробки. Описана організація роботи системних служб. Докладно описано і проілюстровано основні процеси в хмарі, в т.ч. міграція існуючих ВМ.

У четвертому розділі обґрунтовано економічну доцільність дослідження методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі шляхом розрахунку показників економічної ефективності, що дало змогу протягом приблизно двох років компенсувати витрати на використання запропонованих технічних рішень.

П'ятий розділ присвячено аналізу вимог охорони праці і техніки безпеки при використанні комп'ютерної техніки, оцінено дію ЕМІ на елементи програмного засобу і методи захисту, розглянуто проблему шкідливих і небезпечних умов праці персоналу в ході роботи з програмним засобом і запропоновано варіанти компенсації за роботу у таких умовах та спеціальні засоби захисту.

В шостому розділі проведено аналіз сучасних програмних продуктів для обробки великих масивів екологічної інформації та розглянуто статистичну оцінку техногенних впливів.

ABSTRACT

Methods and tools of virtual machines migration monitoring in a computing cloud // Master thesis // Severyn Dmytro // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Nets, group CIm - 62 // Ternopil, 2019 // p.- , fig. – 24 , table. – 6 , Sheets A1 - 10 , Ref. - 18 .

Keywords: CLOUD, HYPERVISOR, MIGRATION, PARAVIRTUALIZATION, RESOURCES, VIRTUALIZATION

The thesis deals with modern virtualization technologies and tools for managing the migration process of virtual machines in a cloud.

The developed software is a set of system services and user interfaces, run on real and virtual servers of the computing complex, which provides a set of computing resources, ie clouds. These services collect information about the use of different system resources and the corresponding server load, and based on this data, statistics on the use of resources, as well as allow the rapid deployment of virtual machines, automatically created based on a given template, on real servers and centrally manage the migration of existing virtual machines based on Xen hypervisor between real servers.

The first section describes the concept of virtualization, the history of its development, conducted an overview of virtualization technologies. The focus is on the analysis of VM-based virtualization. The main advantages and disadvantages of virtualization are given.

The second section describes the concept of a hypervisor and explores its types. The basic functions and capabilities of the most used hypervisors (Xen, VMware ESX Server, Hyper-V, KVM) are analyzed. The peculiarities of VM migration in a cloud are investigated.

The third section describes the functionality and architecture of the developed Xen Migr software. The software architecture of the development is given. The

organization of work of system services is described. The basic processes in the cloud are described and illustrated in detail. migration of existing VMs.

The fourth section substantiates the economic feasibility of researching methods and tools for managing the migration of virtual machines in the cloud by calculating cost-effectiveness indicators, which made it possible to offset the costs of using the proposed technical solutions for approximately two years.

The fifth section deals with the analysis of occupational safety and health requirements when using computer equipment, evaluates the effect of EMI on software elements and methods of protection, examines the problem of harmful and hazardous working conditions of personnel while working with the software, and offers options for compensation for work in such conditions and special remedies.

The sixth section analyzes modern software products for processing large arrays of environmental information and examines the statistical assessment of man-made impacts.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1 Поняття та розвиток віртуалізації.	13
1.2 Огляд технологій віртуалізації.....	17
1.2.1 Віртуалізація на рівні операційної системи	17
1.2.2 Віртуалізація на основі віртуальних машин.....	18
1.3 Переваги і недоліки віртуалізації	24
1.4 Висновки до розділу	26
РОЗДІЛ 2 ДОСЛІДЖЕННЯ МІГРАЦІЇ ВІРТУАЛЬНИХ МАШИН В ОБЧИСЛЮВАЛЬНІЙ ХМАРІ	29
2.1 Гіпервізори та їх типи	29
2.1.1 XEN.....	33
2.1.2 VMware ESX Server.....	36
2.1.3 Hyper-V	37
2.1.4 KVM.....	38
2.2 Особливості міграції ВМ в хмарі	39
2.2.1 Огляд концепції	39
2.2.2 Етапи процесу проведення міграції.....	41
2.2.3 Проведення міграції в гіпервізорах	41
2.2.4 Про загальні мережеві сховища даних.....	42
2.3. Висновки до розділу	43
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ XEN MIGR ДЛЯ УПРАВЛІННЯ ПРОЦЕСОМ МІГРАЦІЇ ВІРТУАЛЬНИХ МАШИН В ОБЧИСЛЮВАЛЬНІЙ ХМАРІ	44
3.1 Функціональні можливості та архітектура засобу.....	45
3.2 Програмна архітектура засобу	46
3.3 Організація роботи системних служб	47
3.4 Основні процеси в хмарі.....	50
3.4.1 Збір статистики використання ресурсів вузлами хмари	50

3.4.2	Спостереження за станом вузлів хмари.....	56
3.4.3	Розгортання нових екземплярів ВМ на основі шаблону.....	60
3.4.4	Міграція існуючих ВМ	63
3.5	Висновки до розділу	67
РОЗДІЛ 4 ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ		68
4.1.	Розрахунок норм часу на виконання науково-дослідної роботи	68
4.2.	Визначення витрат на оплату праці та відрахувань на соціальні заходи	69
4.3.	Розрахунок матеріальних витрат	72
4.4.	Розрахунок витрат на електроенергію	73
4.5.	Розрахунок суми амортизаційних відрахувань	74
4.6.	Обчислення накладних витрат	74
4.7.	Складання кошторису витрат та визначення собівартості НДР	75
4.8.	Розрахунок ціни науково-дослідної роботи	76
4.9.	Визначення економічної ефективності і терміну окупності капітальних вкладень	77
4.10	Висновки до розділу	78
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ		
5.1.	Охорона праці	79
5.2.	Оцінка дії електромагнітного імпульсу на елементі виробництва і методи захисту.	83
5.3.	Врахування шкідливих і небезпечних умов праці персоналу в ході провадження виробничої діяльності суб'єктами господарювання.	86
5.4.	Висновки до розділу	88
РОЗДІЛ 6 ЕКОЛОГІЯ		89
6.1.	Аналіз сучасних програмних продуктів для обробки великих масивів екологічної інформації	89
6.2.	Статистична оцінка техногенних впливів	91
6.3.	Висновки до розділу	94
ВИСНОВКИ.....		95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		96
Додаток А. Тези конференції		98
Додаток Б. Посібник користувача		101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

GNU (General Public License) – загальна публічна ліцензія

HTTP (Hyper Text Transfer Protocol) – протокол передачі гіпер-текстових документів

NAS (Network Attached Storage) – мережеве сховище

NFS (Network File System) – мережева файлова система

SAN (Storage Area Network) – мережа зберігання даних

SQL (Structured query language) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними базами даних

SSH (Secure SHell) – безпечна оболонка

XML (eXtensible Markup Language) – запропонований консорціумом WorldWideWeb (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.

БД – база даних

ВМ – віртуальна машина

ОС – операційна система

СВМ – сервер віртуальної машини

СКБД – система керування базами даних

ПЗ- програмне забезпечення

ВСТУП

Актуальність теми. Одним з сучасних підходів до організації корпоративної інформаційної інфраструктури є віртуалізація обчислювальних систем. Причин цього чимало, але серед основних можна назвати наступні:

– різноманітність платформ, що використовуються в бізнесі (можливість максимально гнучкого використання обчислювальних потужностей для вирішення самого широкого кола завдань неможливо переоцінити);

– проблема простоїв і неефективного використання обладнання (навантаження обчислювальної системи нерідко розподіляється в часі нерівномірно, що призводить до довгих періодів низького навантаження, що становить більшу частину роботи системи, і періодів, коли спостерігаються різкі сплески навантаження);

– низька щільність обчислювальних процесів на одиницю апаратури (проблема також пов'язана з неоптимальним використанням обчислювальних ресурсів обладнання: при виконанні на одному фізичному обчислювачі декількох незалежних призначених для користувача завдань ефективність використання ресурсів підвищується, знімаючи також необхідність в окремому обчислювачі, виділеному для виконання того чи іншого користувацького завдання).

Технологія віртуалізації дозволяє значно покращити процес побудови хмарних сервісів, оскільки зменшує кількість фізичного обладнання, вартість накупівлю і обслуговування якого може бути досить високою. Варто зауважити, що існують проблеми з переходом на хмарні технології, наприклад робота сервісів та програмного забезпечення від різних виробників для різних платформ. Початок використання хмарних сервісів часто гальмується питаннями перенесення інфраструктури або окремих її частин у хмару. Для вирішення цих проблем застосовуються технологія міграції. Необхідно забезпечити грамотне керування процесом міграції, завдяки чому можна буде переносити практично будь-яке навантаження x86 у хмару.

Зв'язок із науковими програмами, планами, темами. Магістерська робота виконана відповідно до наукової тематики Тернопільського національного

технічного університету імені Івана Пулюя, кафедри комп'ютерних систем та мереж.

Мета роботи: дослідити існуючі методи та інструментальні засоби і на їх основі розробити програмний засіб для управління процесом міграції VM в обчислювальній хмарі.

Об'єкт дослідження: процес міграції віртуальних машин в обчислювальній хмарі.

Предмет дослідження: технології віртуалізації комп'ютерних систем, мережеві технології.

В роботі поставлено та розв'язано **наступні задачі:**

- проаналізувати сучасні методиками і технології віртуалізації;
- дослідити основні можливості сучасних гіпервізорів;
- дослідити особливості міграції VM в обчислювальній хмарі;
- розробити програмний засіб для забезпечення можливостей централізованого управління міграцією VM між вузлами хмари.

Наукова новизна отриманих результатів.

- запропоновано архітектуру засобу для проведення міграції, яка складається з одного сервера-координатора та довільної кількості серверів-агентів;
- запропоновано метод проведення міграції віртуальних машин без наявності спільного мережевого сховища даних;
- спроектовано та реалізовано програмний засіб, що складається з реальних і віртуальних обчислювачів, що працюють під управлінням системи віртуалізації обчислювальних ресурсів.

Методи дослідження: Метод теоретичного дослідження та експериментальний з використання персонального комп'ютера. Методологічну основу дослідження становлять фундаментальні положення комп'ютерної на програмної інженерії, наукові дослідження вітчизняних і зарубіжних компаній та вчених у сфері комп'ютеризованих систем.

Практичне значення одержаних результатів. Впровадження результатів проведеного дослідження можуть бути використані компаніями при побудові

власного корпоративного хмарного середовища, в навчальних дисциплінах при розгляді технологій віртуалізації та хмарних обчислень.

Публікації. Результати дослідження доповідалися на VII науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (11-12 грудня 2019 р.) у вигляді тез:

Северин Д.А. Програмний засіб для управління процесом міграції віртуальних машин в обчислювальній хмарі. Інформаційні моделі, системи та технології: Праці VII наук.-техн. конф. (Тернопіль, 11-12 грудня 2019 р.) Тернопіль, 2019. С. 96.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 6 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 101 арк. формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття та розвиток віртуалізації

Віртуалізація - це створення ізольованих оточень в рамках одного фізичного пристрою (комп'ютера). Кожне оточення при цьому виглядає як окремий комп'ютер зі своїми характеристиками, такими як пам'ять, що процесор тощо. Таке оточення називають набором логічних ресурсів або VM. Віртуалізація дозволяє запускати ОС як звичайну програму на комп'ютері.

Віртуалізація дозволяє оптимізувати завантаження систем в періоди простою, за рахунок чого обчислювальні потужності використовуються більш економно, а ефективність діяльності підприємства підвищується. Зараз можливість запуску декількох VM на одній фізичній викликає великий інтерес серед комп'ютерних фахівців, не тільки тому, що це підвищує гнучкість ІТ-інфраструктури, а й тому, що віртуалізація, насправді, дозволяє економити гроші.

В основі віртуалізації лежить можливість одного комп'ютера виконувати роботу декількох комп'ютерів завдяки розподілу його ресурсів за кількома середовищами (рис. 1.1).

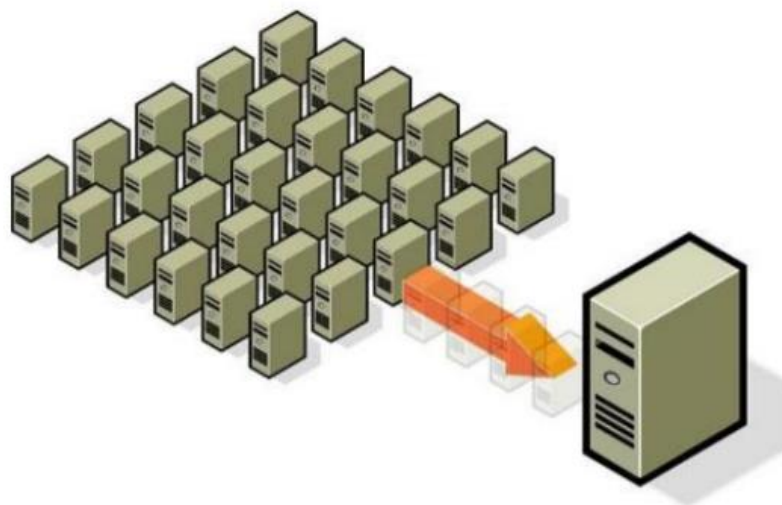


Рис 1.1. Віртуалізація (запуск на одному фізичному комп'ютері декількох віртуальних комп'ютерів)

За допомогою віртуальних серверів і віртуальних настільних комп'ютерів можна розмістити кілька ОС і кілька додатків в єдиному місці розташування. Отже, фізичні і географічні обмеження перестають мати будь-яке значення.

Крім енергозбереження та скорочення витрат завдяки більш ефективному використанню апаратних ресурсів, віртуальна інфраструктура забезпечує високий рівень доступності ресурсів, більш ефективну систему управління, підвищену безпеку і вдосконалену систему відновлення в критичних ситуаціях. [1]

Таким чином, перехід від фізичного відокремлення серверів до логічного – це також є віртуалізацією (рис. 1.2). Віртуалізація ресурсів фізичного сервера дозволяє гнучко розподіляти їх між додатками, кожен з яких при цьому "бачить" тільки призначені йому ресурси і "вважає", що йому виділено окремий сервер, тобто в даному випадку реалізується підхід "один сервер - кілька додатків", але без зниження продуктивності, доступності та безпеки серверних додатків.

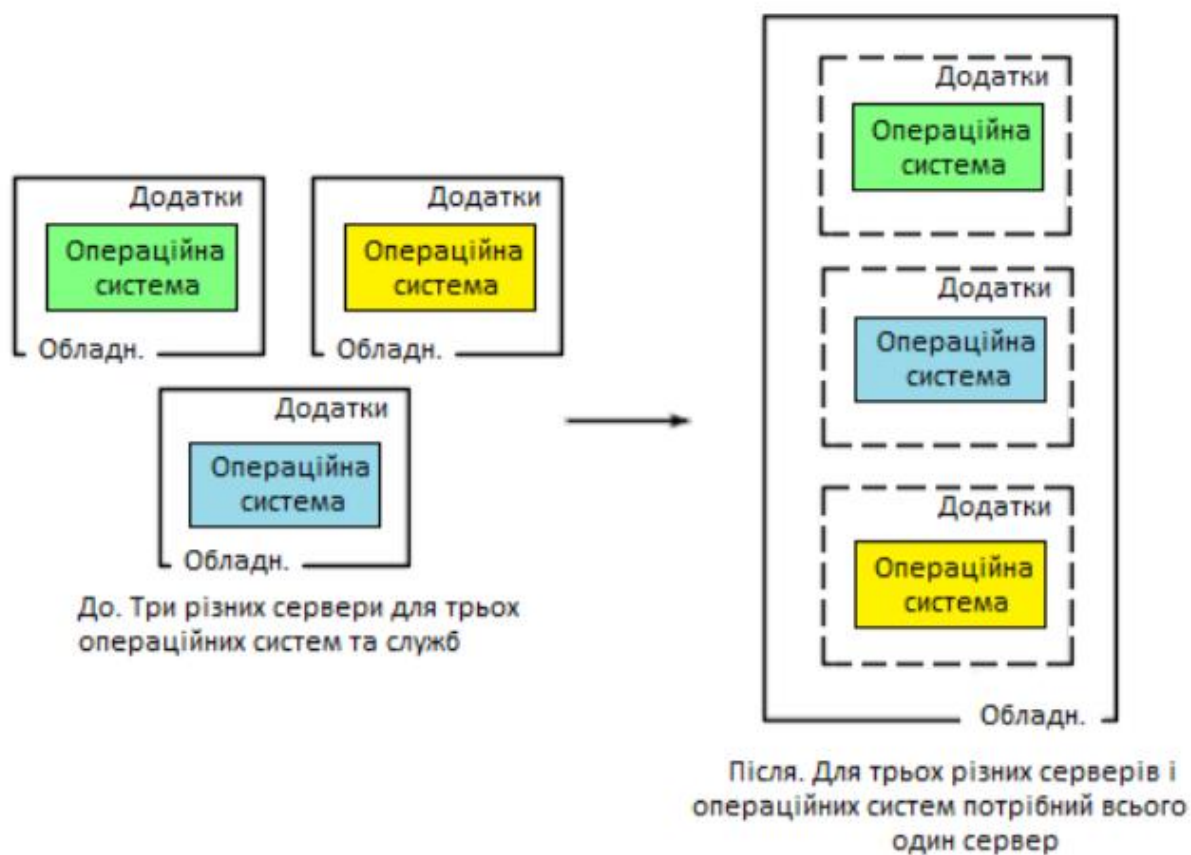


Рис 1.2. Перехід від фізичного відокремлення серверів до логічного (віртуалізація)

Крім того, рішення віртуалізації дають можливість запускати в розділах різні ОС за допомогою емуляції їх системних викликів до апаратних ресурсів сервера.

У широкому сенсі, поняття віртуалізації являє собою приховування справжньої реалізації будь-якого процесу або об'єкта від істинного його представлення для того, хто ним користується. Продуктом віртуалізації є щось зручне для використання і яке насправді, має більш складну або зовсім іншу структуру, відмінну від тієї, яка сприймається при роботі з об'єктом. Іншими словами, відбувається відділення представлення від реалізації чого-небудь. Віртуалізація покликана абстрагувати ПЗ від апаратної частини.

Вперше технології віртуалізації були розроблені і застосовані компанією ІВМ ще в 1960-х рр. на обчислювальних комплексах, названих мейнфреймами. Тоді віртуалізація представляла чисто науковий інтерес і була оригінальним рішенням для ізоляції комп'ютерних систем в рамках одного фізичного комп'ютера.

Необхідність створення була обумовлена тим, що обчислювальна система була достатньо дорогим і рідкісним ресурсом, а тому для задоволення потреб всіх користувачів системи був потрібний механізм спільного використання обчислювальних ресурсів єдиної системи. Так народилася технологія поділу обчислювальних потужностей між користувачами шляхом надання кожному з них відокремленої логічної середовища - так званої ВМ. Обчислювальні процеси користувачів були відокремлені один від одного, і віртуально кожен з них був одноосібним користувачем системи. В результаті основна обчислювальна система, тобто власне мейнфрейм, використовувалася досить ефективно, а віддача від інвестицій в обчислювальні комплекси була максимальною.

Популярність технологій віртуалізації продовжувала зростати приблизно до 1980-х рр., коли над архітектурою обчислювальних комплексів, заснованих на основному комплекті, стала переважати архітектура «клієнт-сервер». Це пов'язано з ростом в той час популярності архітектури процесора x86, з самого початку не призначеної для задач віртуалізації, та здешевленням комп'ютерів і серверів на основі процесорів цієї архітектури.

Однак зі зростанням продуктивності і технологічними вдосконаленнями процесорів проблема нераціонального використання ресурсів повернулася знову. Як стверджується в [2], більшість серверів в середньому задіють в роботі всього 15-19% від загальної кількості ресурсів. Низька ефективність використання серверів пояснюється широко застосовуваним з початку 90-х років підходом "один додаток - один сервер", тобто кожного разу для розгортання нової програми компанія купує новий сервер. Очевидно, що на практиці це означає швидке збільшення серверного парку і, як наслідок - зростання витрат на його адміністрування, енергоспоживання та охолодження, а також потреби в додаткових приміщеннях для установки все нових серверів і придбання ліцензій на серверну ОС. Таким чином, ідеї віртуалізації знову стали досить затребуваними.

Перші сучасні розробки в цій області були пов'язані з подоланням обмежень не призначеної для віртуалізації архітектури x86, а пізніше з'явилися й процесори з підтримкою технологій апаратної віртуалізації – як для клієнтських комп'ютерів, так і для серверів.

В 1999 р американська компанія VMware представила технологію віртуалізації систем на базі x86 в якості ефективного засобу, здатного перетворити такі системи в єдину апаратну інфраструктуру загального користування та призначення, що забезпечує повну ізоляцію, мобільність і широкий вибір ОС для прикладних середовищ. [2] Компанія VMware була однією з перших, хто зробив серйозну ставку виключно на віртуалізацію. В подальшому це виявилось абсолютно виправданим.

Сьогодні VMware пропонує комплексну віртуалізаційну платформу VMware vSphere, яка включає засоби як для окремого ПК, так і для центру обробки даних. Ключовим компонентом цього програмного комплексу є гіпервізор VMware ESX Server. Пізніше в "битву" за місце в цьому модному напрямку розвитку інформаційних технологій включилися такі компанії як Parallels, Oracle, Citrix Systems.

Корпорація Microsoft вийшла на ринок засобів віртуалізації в 2003 році з придбанням компанії Connectix, випустивши свій перший продукт Virtual PC для настільних ПК. З тих пір вона послідовно нарощувала спектр пропозицій в цій

галузі і на сьогодні майже завершила формування віртуалізаційної платформи, до складу якої входять такі рішення як Windows Server R2 з компонентом Hyper-V, Microsoft Application Virtualization (App-v), Microsoft Virtual Desktop Infrastructure (VDI), Remote Desktop Services, System Center Virtual Machine Manager.

1.2. Огляд технологій віртуалізації

На даний час способи віртуалізації комп'ютерних систем можна об'єднати в дві групи, які нижче будуть розглянуті докладніше:

- віртуалізація на рівні ОС (часто зустрічаються назви контейнеризація або контейнерна віртуалізація);
- віртуалізація на основі ВМ. [3]

1.2.1. Віртуалізація на рівні операційної системи

Цим терміном позначають спосіб організації роботи системи, при якому на урахуванням одного екземпляру ядра ОС створюється безліч логічно незалежних системних користувальницьких оточень (див. рис. 1.3). Ця концепція заснована на ідеї підміни кореневого каталогу (chroot) - функції, яка існує у всіх сучасних UNIX-подібних ОС.



Рис.1.3. Віртуалізація на рівні ОС

На етапі завантаження системи ядро може використовуватися як кореневу файловою систему так звані фантомні диски, тобто області оперативної пам'яті, змонтовані так, ніби вони були віртуальними носіями, для завантаження драйверів і виконання інших низькорівневих завдань, пов'язаних із запуском системи. Потім ядро може змінити кореневий каталог за допомогою команди `chroot`, щоб змонтувати певну дискову файловою систему в якості кореневої, і продовжити завантаження з неї. Механізм підміни кореневого каталогу в тому вигляді, в якому він використовується за віртуалізації рівня ОС, розширює цю ідею, дозволяючи ОС запускати віртуальні сервери - кожен зі своїм набором процесів, які виконуються щодо їх власних корневих каталогів. Будучи обмеженими своїми корневими каталогами та відповідними файловими системами, віртуальні сервери не можуть отримати доступу до файлів один одного - таким чином забезпечується базовий захист від атак на процеси віртуального сервера і нього самого. Навіть якщо сервер з підміненим корневим каталогом стане об'єктом вдалої атаки, у нього є доступ тільки до файлів в межах його власної кореневої файлової системи.

Переваги - висока ефективність використання апаратних ресурсів (автономна організація; малий розмір; мале споживання пам'яті), низькі накладні технічні витрати, відмінна керованість, мінімізація витрат на придбання ліцензій. Недоліки - реалізація тільки однорідних обчислювальних середовищ (вся суть контейнера полягає в запуску одного ізольованого додатку); безпека використання.

Найбільш відомими прикладами реалізації цієї концепції є технології FreeBSD Jails, OpenVZ, LXC, Solaris Containers, iCore Virtual Accounts, Docker.

Далі в дипломній роботі ця концепція розглядатися не буде.

1.2.2. Віртуалізація на основі віртуальних машин

ВМ прийнято називати якимось ізольоване програмне або апаратне середовище виконання деяких обчислювальних процесів. ВМ представляє собою ОС, яка має власне ядро, драйвери для роботи з фізичним обладнанням, додатки

користувача, власні налаштування як і у звичайних ОС. VM не може самостійно виконувати обчислювальні процеси та взаємодіяти з фізичними пристроями. Для виконання VM користувальницьких завдань та забезпечення її функціонування фактично використовуються обчислювальні ресурси реальної системи, в середовищі якої запущена ця VM. VM є всього лише набором програм, які виконуються в системі у виділеній на період існування машини області оперативної пам'яті. VM можуть бути реалізовані по-різному, і список їх частих способів застосування включає в себе наступні варіанти:

- захист інформації;
- емуляція різних процесорних архітектур;
- оптимізація використання ресурсів фізичних обчислювачів;
- моделювання клієнт-серверних систем на єдиному обчислювачі.

VM діє так само, як фізичний комп'ютер, і містить власні віртуальні (тобто програмні) ОЗП, жорсткий диск і мережевий адаптер. ОС не може розрізнити віртуальну і фізичну машини. Те ж саме можна сказати про додатки та інших комп'ютерах в мережі. Навіть сама VM вважає себе "справжнім" комп'ютером. Але незважаючи на це VM складаються виключно з програмних компонентів і не включають обладнання.

VM притаманні наступні особливості: [3]

- сумісність з усіма стандартними комп'ютерами, ОС, програмами та драйверами пристроїв (як і фізичний комп'ютер, VM працює під управлінням власної гостьової ОС і виконує власні додатки. Вона також містить всі компоненти, стандартні для фізичного комп'ютера (материнську плату, відеокарту, мережевий контролер і т. д.). VM можна використовувати для виконання будь-якого ПЗ, придатного для відповідного фізичного комп'ютера);
- повна ізоляваність однієї VM від іншої (VM можуть використовувати загальні фізичні ресурси одного комп'ютера і при цьому залишатися повністю ізольованими один від одного, як якщо б вони були окремими фізичними машинами. Наприклад, якщо на одному фізичному сервері запущено чотири VM, і одна з них дає збій, це не впливає на доступність інших трьох машин);
- інкапсуляція (VM повністю інкапсулюють обчислювальне середовище, завдяки чому VM можна перемістити або скопіювати з одного місця до іншого як

будь-який інший програмний файл. Крім того, VM можна зберегти на будь-якому стандартному носії даних: від компактної карти Flash-пам'яті USB до корпоративних мереж зберігання даних);

– незалежність від обладнання (VM повністю незалежні від базового фізичного обладнання, на якому вони працюють. Наприклад, для VM з віртуальними компонентами (ЦП, мережевою картою, контролером SCSI) можна задати налаштування, що абсолютно не збігаються з фізичними характеристиками базового апаратного забезпечення. VM можуть навіть виконувати різні ОС (Windows, Linux і ін.) на одному і тому ж фізичному сервері).

У контексті даної роботи під VM слід розуміти програмний контейнер, здатний імітувати роботу окремого комп'ютера, що має свою власну ОС і набір додатків, а також системні пристрої - такі як, наприклад, жорсткий диск, мережевий контролер, процесор або оперативна пам'ять.

Цей віртуальний комп'ютер за замовчуванням повністю ізольований від зовнішнього середовища і інших віртуальних комп'ютерів, імітація роботи його пристроїв досягається виключно за рахунок відповідного ПЗ. Для виконання VM завдань користувача і забезпечення функціонування її ОС фактично використовуються обчислювальні ресурси реальної системи, в середовищі якої запущена ця VM - адже власне кажучи VM є всього лише набір виконуються в системі програми і виділена на період існування машини область оперативної пам'яті, з точки зору машини знаходиться під повним і неподільним контролем останньої.

Щоб бути точнішим, необхідно зробити кілька застережень стосовно суто програмної природи VM і їх повної ізоляції від зовнішнього світу.

По-перше, різні інструменти віртуалізації дозволяють VM отримувати прямий доступ до деяких призначених для користувача пристроїв - наприклад, USB-накопичувачів, розділах жорстких дисків або навіть цілою диском.

По-друге, VM нерідко включаються в існуючі комп'ютерні мережі в якості повноцінних членів або навіть особливі віртуальні мережі, що складаються виключно з VM, чия робота фізично відбувається тільки в оперативній пам'яті фізичного обчислювача.

По-третє, процесори останніх поколінь надають спеціальні машинні команди і структури даних для створення VM і управління ними.

Однак слід розуміти, що без налаштованої явно комунікації із зовнішнім світом - за допомогою мережі, загального дискового простору або чого-небудь ще - VM повністю замкнута у виділеній їй області пам'яті і не може отримати доступу до процесів або даними як фізичного обчислювача, так і інших VM, адресний простір її оперативної пам'яті і кордони відведеного для її роботи дискового простору строго обмежені, незважаючи на те, що фізично ця машина ділить з собі подібними і фізичним обчислювачем якісь єдині ресурси.

В даний час існують наступні типи віртуалізації на основі VM:

- гостьова ОС (англ. Guest OS);
- паралельна VM (англ. Parallel virtual machine);
- на основі гіпервізора (англ. Hypervisor-based): паравіртуалізація (англ. Paravirtualization) та повна віртуалізація (англ. Full virtualization);
- віртуалізація на рівні ядра (англ. Kernel-level virtualization).

Розглянемо всі їх докладніше

Гостьова ОС. Кожна VM виконується як окремий екземпляр ОС всередині програми, що відповідає за віртуалізацію, яке, в свою чергу, працює під управлінням конкретної ОС. Приклади реалізації цього підходу - Parallels Workstation, VMWare Workstation, Microsoft VirtualPC, Oracle VirtualBox. ОС, під керуванням якої працює ПЗ віртуалізації, часто називають хост-системою (від англ. host - букв. «господар»), оскільки вона надає додаткові, що забезпечує віртуалізацію, середовище виконання.

Паралельна VM. Кілька реальних або віртуальних обчислювачів утворюють єдину VM за допомогою засобів кластеризації - наприклад, пакету Parallel Virtual Machine (PVM). Отриманий в результаті кластер здатний виконувати складні обчислення і операції, пов'язані з інтенсивною роботою з даними на дисках, за рахунок кооперації складових його обчислювачів.

Віртуалізація на основі гіпервізора. При використанні цього підходу в архітектурі системи з'являється додатковий елемент - так званий гіпервізор, що представляє собою легко організований і високо оптимізований програмний прошарок між апаратним та ПЗ. У термінах концепції кілець захисту (див. рис.

1.4) гіпервізор виконується в нульовому кільці (англ. ring 0) - хоча зазвичай з таким рівнем привілеїв виконується ОС. У свою чергу, ОС реального обчислювача і ВМ виконуються в менш привілейованих кільцях. [4]

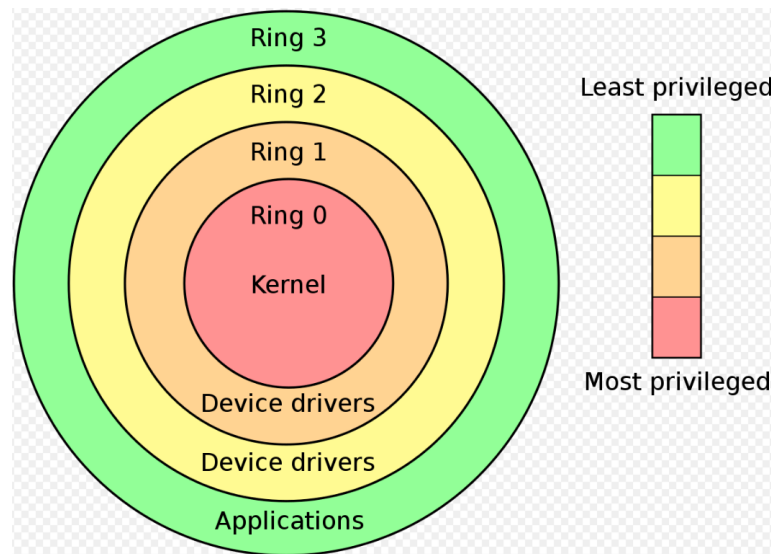


Рис.1.4. Кільця захисту (рівні привілеїв гіпервізора)

Гіпервізор виконує кілька основних завдань:

– здатний розпізнавати, перехоплювати і відповідати на так звані привілейовані команди - особливий клас машинних команд, які можуть бути породжені тільки ядром ОС. Ці команди потім емуляються гіпервізором, як якщо б вони були виконані від імені відповідної ОС - цей прийом називається зниженням рівня привілеїв кільця (англ. Ring deprivileging [5]).

– відповідає за впорядкування, перенаправлення і повернення результатів запитів від ОС до обладнання. Керуюча ОС виконується у середовищі гіпервізора - так само, як і всі ВМ, вона використовується для управління гіпервізором і ВМ.

Таким чином, гіпервізор розширює ідею спільного використання ресурсів безліччю призначених для користувача завдань і вирішує завдання планування процесорного часу подібно до того, як це робить будь-яка ОС - тільки для гіпервізора споживачами ресурсів є не окремі додатки, а цілі ОС, що, звичайно ж, ускладнює завдання.

Розгляньмо тепер підтипи віртуалізації на основі гіпервізора - паравіртуалізацію і повну віртуалізацію

Паравіртуалізація (рис. 1.5). Використання цього підходу вимагає внесення змін в ядро ОС, яка буде працювати в середовищі гіпервізора. Використовуючи паравіртуалізацію, можна домогтися достатньо великого виграшу в продуктивності в порівнянні з використанням інших концепцій, оскільки зміни в ядрі ОС дають змогу їй працювати з гіпервізором безпосередньо, без додаткових накладних витрат, які виникають при використанні інших технік віртуалізації, пов'язаних з емуляцією - як на основі гіпервізора, так і без нього.



Рис.1.5. Паравіртуалізація

Однак, необхідність внесення змін в ядро системи є серйозною перешкодою, що робить неможливим запуск багатьох типів ОС на гіпервізорі. Переваги даної технології полягають у відсутності потреби в хостовій ОС – VM встановлюються фактично на "голе залізо", а апаратні ресурси використовуються ефективно. Недоліки - в складності реалізації підходу і необхідності створення спеціалізованої ОС-гіпервізора. Основні приклади продуктів, що надають можливості віртуалізації - Microsoft Hyper-V, Xen, VMWare ESX Server.

Повна віртуалізація. У разі повної віртуалізації гіпервізор включає в себе код, який, при необхідності, дозволяє прозора для VM виробляти емуляцію обладнання - таким чином стає можливим запуск навіть не модифікованих ОС (рис. 1.6).

Таку модель роботи використовує продукт VMWare ESX Server. В Xen також є можливість повної віртуалізації. В обох випадках підтримка апаратної віртуалізації процесором реального обчислювача є обов'язковою вимогою.

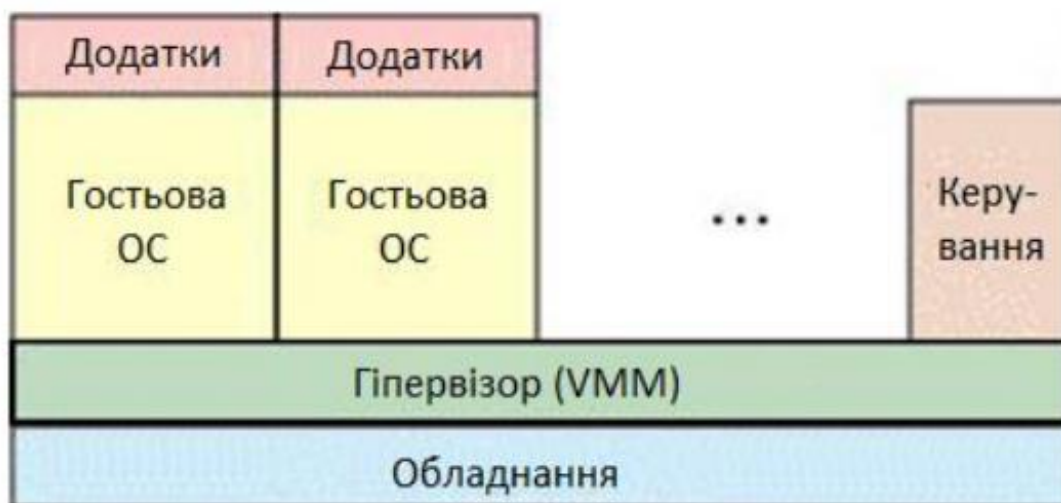


Рис.1.6. Повна віртуалізація

До переваг даного підходу можна зарахувати відносну простоту реалізації, універсальність і надійність рішення; всі функції управління бере на себе хост-ОС. Недоліки - високі додаткові накладні витрати на апаратні ресурси, відсутність врахування особливостей гостьових ОС, менша, ніж потрібно, гнучкість у використанні апаратних засобів.

Віртуалізація на рівні ядра. При використанні цього підходу гіпервізор не потрібен - натомість запускається окремий екземпляр ядра Linux, який разом із пов'язаною з ним ВМ виконується як звичайний користувальницький процес. Таким чином забезпечується проста можливість запускати безліч ВМ на одному реальному обчислювачі.

Типовими прикладами реалізацію даного підходу є User-Mode Linux (UML) і KVM (Kernel Virtual Machine), для роботи останнього необхідна підтримка процесором апаратної віртуалізації.

1.3. Переваги і недоліки віртуалізації

На основі проведеного огляду, можна стверджувати, що основними перевагами технологій традиційної віртуалізації є:

- ефективне використання обчислювальних ресурсів (замість трьох, а то і десятих серверів, завантажених на 5 -20 % можна використовувати один, завантажений на 50 -70 %);

- скорочення витрат на інфраструктуру (таким чином потреби в обслуговуванні, електроживленні й охолодженні матеріальних ресурсів скорочуються, і на ІТ витрачається значно менше коштів);

- зниження витрат на ПЗ (купуючи одну ліцензію на Microsoft Windows Server 2008 Enterprise, отримуєте право одночасно її використовувати на 1 фізичному сервері і 4 віртуальних (в межах одного сервера), а Windows Server 2008 Datacenter ліцензується тільки на кількість процесорів і може використовуватися одночасно на необмеженій кількості віртуальних серверів);

- підвищення гнучкості і швидкості реагування системи (при використанні віртуального сервера – можливий моментальний запуск на будь -якому залізі, а якщо немає подібного сервера, то можна скачати готову ВМ з встановленим та налаштованим сервером, з бібліотек, підтримуваних компаніями розробниками гіпервізорів (програм для віртуалізації);

- можливість роботи на одному комп'ютері несумісних додатків (при використанні віртуалізації на одному сервері можлива установка Linux і Windows серверів, шлюзів, баз даних і інших абсолютно несумісних в рамках однієї не віртуалізованої системи додатків);

- підвищення доступності додатків і забезпечення безперервності роботи підприємства (равдяки надійній системі резервного копіювання та міграції віртуальних середовищ цілком без перерв в обслуговуванні ви зможете скоротити періоди планового простою і забезпечити швидке відновлення системи в критичних ситуаціях. "Падіння» одного віртуального сервера не веде до втрати інших віртуальних серверів. Крім того, у разі відмови одного фізичного сервера можливо зробити автоматичну заміну на резервний сервер);

– можливість легкої архівації (оскільки жорсткий диск VM, зазвичай, представляється у вигляді файлу певного формату, розташований на якому - небудь фізичному носії, віртуалізація дає можливість простого копіювання цього файлу на резервний носій як засіб архівування і резервного копіювання всієї VM цілком. Можливо підняти сервер з архіву, не знищуючи поточний сервер і подивитися стан справ за минулий період);

– підвищення керованості інфраструктури (використання централізованого управління віртуальною інфраструктурою дозволяє скоротити час на адміністрування серверів, забезпечує балансування навантаження і "живу" міграцію VM).

Але технологія віртуалізації має і певні недоліки:

– в процесі реалізації деяких проектів можуть виникати ситуації, коли засоби віртуалізації не дозволяють прикладному ПЗ або інформаційним системам клієнта працювати з периферійними пристроями (потрібно зібрати набір необхідного периферійного обладнання, пакет прикладного ПЗ, реалізувати заплановану архітектуру і домогтися повноцінної роботи і функціональності тестованого рішення);

– витрата обчислювальної потужності сервера на віртуалізацію (щоб віртуалізація була ефективною, потрібно дуже уважно стежити за завантаженням віртуальних серверів і своєчасним виділенням достатніх для оптимальної роботи ресурсів. Всі віртуальні сервери необхідно ретельно налаштовувати, для чого зазвичай використовуються засоби моніторингу навантаження, якими оснащуються всі віртуальні сервери);

– застосування засобів віртуалізації, як правило, передбачає використання як мінімум двох комплектів апаратних засобів (таке рішення корисно ще й тим, що позбавляє від збоїв апаратури. Таким чином, під час налаштування віртуальних серверів необхідно застосовувати спеціальні засоби визначення завантаженості, що допомагають встановлювати оптимальний баланс навантаження і, крім того, рекомендується використовувати такі архітектурні рішення засобів віртуалізації, які встановлюються на порожню апаратну платформу).

1.4. Висновки до розділу

У цьому розділі описано поняття віртуалізації, історія її розвитку, проведено огляд технологій віртуалізації. Основну увагу приділено аналізу процесу віртуалізації на основі ВМ. Наведено основні переваги та недоліки віртуалізації.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ МІГРАЦІЇ ВІРТУАЛЬНИХ МАШИН В ОБЧИСЛЮВАЛЬНІЙ ХМАРІ

2.1. Гіпервізори та їх типи

Гіпервізор - це вбудоване ПЗ чи програма, яка робить можливим віртуалізацію системних ресурсів і дозволяє паралельну роботу кількох ОС на одній машині [8]. Основна ідея використання гіпервізора – забезпечити функціонування середовищ виконання (ізольованих) для кожної ВМ та керування доступом ВМ і гостьової ОС до апаратних ресурсів комп'ютера. Гіпервізор створює з одного фізичного комп'ютера кілька копій, клонів його апаратних ресурсів, і кожен клон видно з боку користувача як окремий пристрій. На кожен ВМ можна встановити гостьову ОС користувача, не прив'язану до «заліза» хоста.

Гіпервізор ізолює запущені ОС одну від одної так, щоб кожна з них монополює використовувала виділені їй ресурси. Але, при необхідності, гіпервізор дозволяє ОС ВМ взаємодіяти між собою. Механізмом зв'язку між ОС може бути загальний доступ до певних файлів і обмін даними по локальній мережі. Таким чином, замість одного комп'ютера як ніби виходить кілька, і кожен з них працює зі своїм ПО незалежно від інших. Однак, в реальності, втілення кожної такої ВМ - лише набір файлів в пам'яті хоста. Зрозуміло, якщо вимкнути фізичний сервер - вся ілюзія тут же зникне, тому що перестане працювати гіпервізор.

Виділяють кілька типів гіпервізорів. [9]

Гіпервізор 1-го типу (тонкий гіпервізор або мікроядро) виконується як контрольна програма безпосередньо на стороні апаратної частини комп'ютера (див. рис. 2.1). ОС ВМ виконуються рівнем вище. Такий гіпервізор найпростіше сприймати як специфічну компактну ОС, яка встановлюється прямо на «залізо» (bare-metal server) і має основні ознаки ОС. Оскільки даний гіпервізор працює незалежно від ОС, він забезпечує більшу продуктивність, надійність і безпеку.

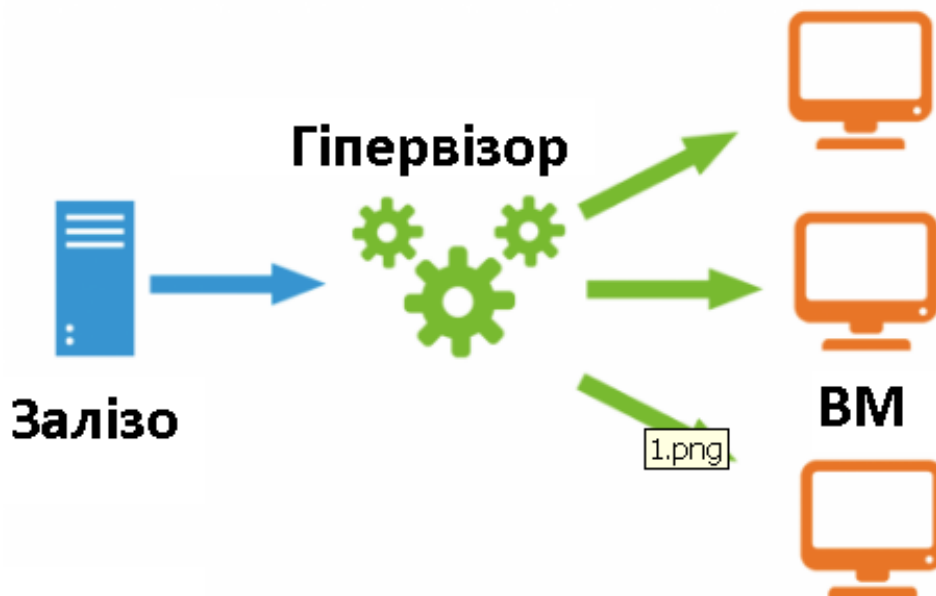


Рис. 2.1. Гіпервізор першого типу

ПЗ гіпервізора такого типу має особливість - розмір його коду в сотні разів менше, ніж у більшості існуючих ОС. Це дозволяє у сотні разів зменшити кількість можливих помилок, що призводять до зависання всієї системи. Збій в роботі ОС на одній з VM користувача не впливає на роботу всіх інших машин. Важливою вимогою до гіпервізора є безпека. Це досягається за рахунок передачі гіпервізору повного управління апаратними ресурсами комп'ютера. Гіпервізор створює «ілюзію» для гостьової ОС, що її код виконується хостом на рівні заліза, в привілейованому режимі, хоча де-факто вона функціонує в режимі користувача. Якщо станеться крах однієї з гостьових систем, робота інших триватиме. Для 1-го типу можна виділяють два окремі підтипи: монолітний та мікроядерний [10].

Монолітний – використовує розміщення гіпервізора в одному рівні з більшістю потрібних компонент, зокрема ядро, драйвери пристроїв та ін (рис. 2.2). Приклади: VMware ESX, Xen.

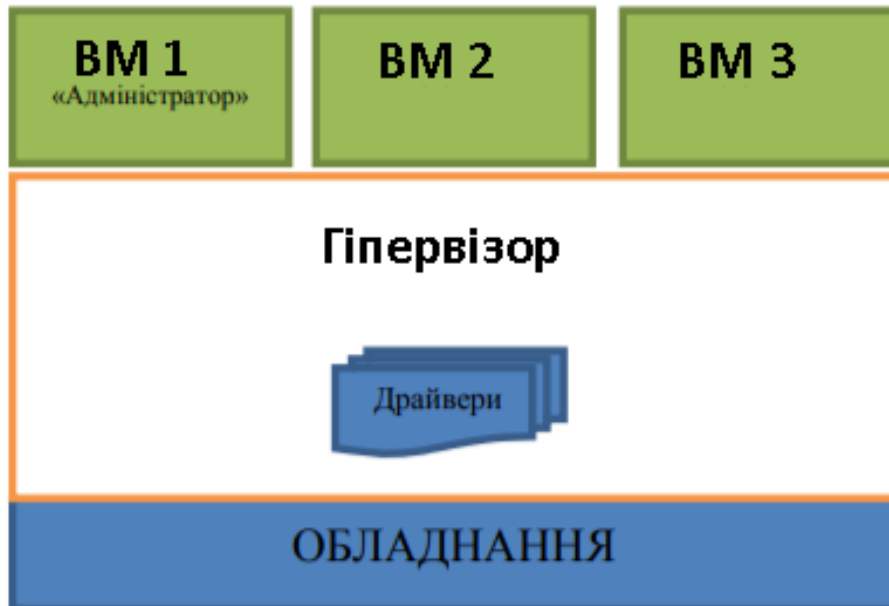


Рис. 2.2. Монолітний гіпервізор

Ця модель має власні драйвери для забезпечення доступу до обладнання. Коли гостьова система потребує такого доступу, тоді вона має скористатися гіпервізором з його моделлю драйверів. Як правило, одна з гостьових ОС є адміністратором або консоллю, в якій необхідно завантажити компоненти для отримання ресурсів, керування і моніторингу всіх гостьових ОС комп'ютера.

Монолітний гіпервізор має хорошу продуктивність, але його слабким місцем є захищеність і стійкість. Така модель гіпервізора дозволяє роботу драйверів (навіть інколи і програм сторонніх виробників). Нехай, шкідлива програма може встановити в гіпервізор реєстратор натискань клавіш під виглядом драйвера пристрою. Наслідком цього буде те, що під контролем зловмисної програми будуть всі гостьові ОС системи. Ще гірше, цей реєстратор абсолютно неможливо буде виявити за допомогою гостьових ОС, оскільки гіпервізор для них невидимий!

Підтип мікроядерий працює з «тонким», спеціалізованим гіпервізором. Виконує тільки основні моменти забезпечення ізолюваності розділів та керування пам'яттю (рис. 2.3). Не містить стеку вводу/виводу та драйверів пристроїв (як приклад, Nucleus-V). Будь-яка гостьова ОС має можливість для отримання доступу до обладнання через гіпервізор. Таким чином кожна VM займає абсолютно окремий розділ. Це якнайкраще впливає на захищеність і надійність.

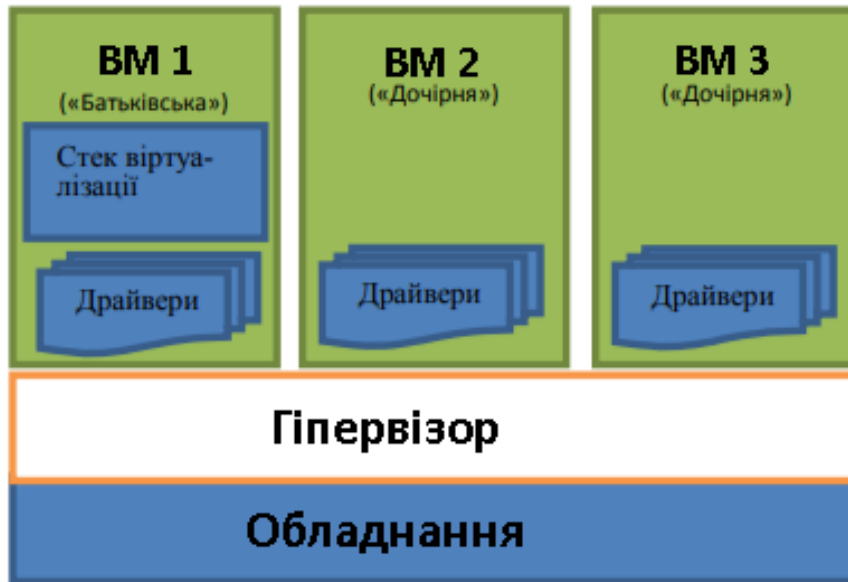


Рис. 2.3. Мікроядерний гіпервізор

Такий тип гіпервізора [9] має істотні переваги відносно монолітного: сумісний з будь-яким обладнанням, драйвери якого розташовуються в рамках батьківської ОС; вищий рівень безпеки; вища продуктивність гіпервізора, оскільки відсутня взаємодія з драйверами пристроїв. Такий тип може бути у незначному програді у продуктивності відносно монолітного. Проте, зараз головним питанням є безпека, тому не є проблемою втрата кількох відсотків продуктивності заради значного підвищення стійкості.

Гіпервізор другого типу виконується в рамках хостової ОС (див. рис. 2.4). Є додатковим програмним шаром, розташованим по верху основної ОС. Гостьові ОС ВМ розташовуються вище на один рівень. Такий гіпервізор працює як один з процесів, які виконуються основною ОС, найчастіше - Linux.

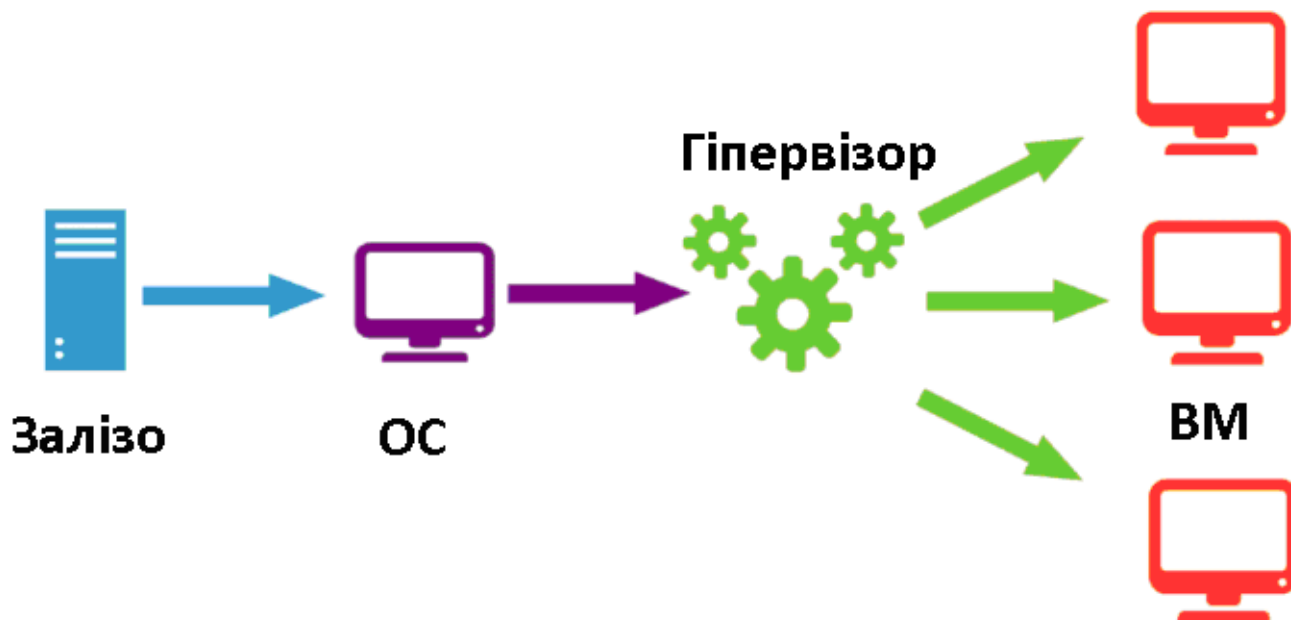


Рис. 2.4. Гіпервізор другого типу

Тут гіпервізор управляє гостьовими ОС, а емуляцію і управління фізичними ресурсами бере на себе хостова ОС. Даний тип має гіршу продуктивність у порівнянні з типом один. Приклади – Oracle VM VirtualBox, VMware Workstation, KVM. Гібридні гіпервізори [9, 10] володіють частиною ознак як першого, так і другого типів (поєднання «тонкого» гіпервізора і спеціальної, яка працює «на залізі», службової ОС під його керуванням). Гіпервізор безпосередньо керує процесором і пам'яттю, а через службову ОС гостьові отримують доступ до пристроїв введення-виведення. Технології постійно розвиваються, і виробники гіпервізора шукають шляхи вдосконалення своїх продуктів, створюють нові версії, більш гнучкі, більш інтегровані до різних систем і умов. В останні роки гіпервізор Xen і Hyper-V все частіше відносять вже не до першого типу, а до гібридного, і частково це вірно. Сучасні версії цих гіпервізорів в значній мірі поєднують в собі властивості обох типів.

2.1.1. XEN

Ідеї, покладені в основу Xen [7], народилися в стінах знаменитого Кембриджського університету, що знаходиться в Англії. Британська історія взагалі налічує чимало прекрасних прикладів творчих та інноваційних розробок, пов'язаних з обчислювальною технікою - варто згадати комп'ютер EDSAC,

«Кембріджське кільце »і багато іншого. Шляхом створення великого пулу обчислювальних ресурсів і механізмів, дозволяють користувачам здійснювати запити ресурсів певного типу, може стати можливим створення і комерційне використання глобальної середовища розподілених обчислень.

Гнучкість використання обчислювальних ресурсів, що дозволяє здійснювати запуск безлічі ОС, дозволяє ефективно здійснювати рішення призначених для користувача завдань шляхом забезпечення спільної роботи безлічі фізичних ресурсів і є наріжним каменем в концепції проекту. І спочатку монітор VM Xen був розроблений для забезпечення цієї функціональності в складі проекту XenServers. Xen надає можливості для управління ресурсами їх обліку та аудиту їхніх використання, а також забезпечує незалежність від конкретної ОС, що є необхідним відповідно до концепції XenServers. Перша доступна публічно версія Xen була випущена в жовтні 2003 року. У цьому ж році була представлена перша доступна широкому загалу стаття про Xen «Xen and the Art of Virtualization» [7].

У 2005 році головні розробники Xen заснували компанію XenSource, Inc., метою якої було розроблення комерційних програмних продуктів на основі Xen для більш зручної організації інфраструктури на основі цієї технології. У 2007 році XenSource, Inc. була поглинена компанією Citrix, яка і є основним розробником технології Xen і продуктів на її основі на сьогоднішній день.

На сьогоднішній день Xen використовується в тисячах комерційних і дослідницьких проектів, має велике співтовариство користувачів і поширюється за ліцензією GNU (GPL), що передбачає відкритий вихідний код проекту. Існує також ряд комерційних рішень, що забезпечують Xen додатковими інструментами для більш зручного використання цього гіпервізора в рішенні задач масштабу підприємства. Перелік компаній, які беруть участь в розробці гіпервізора, включає в себе такі відомі імена, як Intel, AMD, Cisco, HP, IBM, Novell, Red Hat і Oracle.

Xen є гіпервізором VM (див. розділ 1.2.2) і може використовуватися для створення VM як шляхом паравіртуалізації, тому і повної віртуалізації, що дозволяє запускати в середовищі гіпервізора ОС з немодифікованим ядром (за умови підтримки процесором апаратної віртуалізації).

Відповідно до описаної концепції гіпервизора, максимальним рівнем привілеїв виконання машинних команд має сам гіпервизор, тоді як керуюча ОС і ВМ виконуються з більш низьким рівнем привілеїв. І хоча для гіпервизора немає принципової різниці між керуючою ОС і системами ВМ, керуюча система все ж володіє більш високим рівнем привілеїв у порівнянні з ВМ. У термінології Xen будь-яка ОС, що працює в його середовищі, називається доменом. Керуюча ОС називається нульовим доменом (англ. domain 0) або коротко - dom0.

ОС ВМ називаються непривілейованими доменами (Англ. Domain unprivileged) або коротко - domU. З урахуванням вищенаведених позначень легко уявити собі архітектуру вузла, що працює під управлінням Xen, у наступному вигляді (рис. 2.5):

Огляд Xen буде неповним без згадки про два основні способи організації обчислювальної інфраструктури, заснованої на віртуалізації.

По-перше, стандартним рішенням є віртуалізація в масштабах підприємства - в цьому випадку компанія має у власності один або кілька серверів, що використовують можливості віртуалізації, а клієнти використовують їх ресурси.

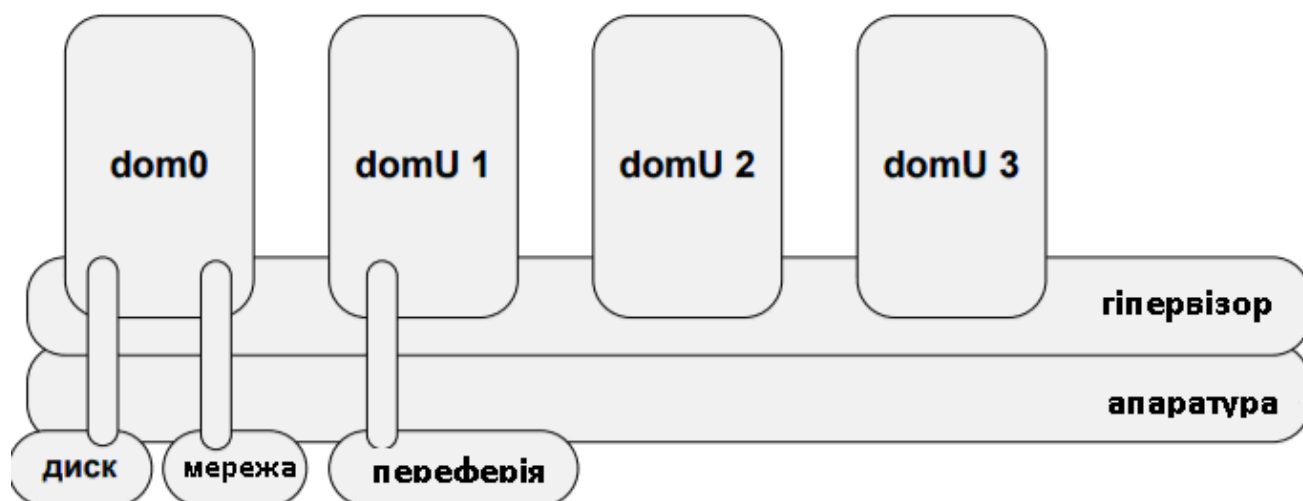


Рис. 2.5. Архітектура Xen-вузла [9]

По-друге, в останні роки набув великої популярності принцип «ПЗ як послуга» (англ. software as a service, SAAS), тісно пов'язаний з концепцією хмарних обчислень. При організації інфраструктури відповідно до принципів цієї концепції формується так звана хмара - великий пул обчислювальних систем, що володіють можливостями віртуалізації, і відповідних системних ресурсів. І підприємства, які відчувають потребу в обчислювальних потужностях, замість

того, щоб купувати сервери у власність, орендують ресурси у компанії, що володіє хмарою. Таким чином, кінцеві споживачі послуг не відчують труднощів із самостійною організацією складної інформаційної інфраструктури та отримують можливість шляхом менших початкових грошових і часових витрат отримати готову корпоративне рішення, а гнучкість використання ресурсів, що забезпечується використанням технологій віртуалізації, дозволяє вирішувати за допомогою хмари завдання самого різного роду.

Неважко здогадатися, що технології на кшталт Xen і подібних їй знаходять широке застосування при організації обчислювальних ресурсів відповідно до описаного вище принципу хмари - якщо говорити конкретно про Xen, то легкість використання в подібних рішеннях пояснюється ще й тим, що початкова концепція XenServers має з хмарою чимало спільного. Xen має в своєму складі кілька технологій, які істотно полегшують втілення в життя принципів хмарної організації ресурсів - однією з головних таких технологій є можливість міграції VM, яка буде розглянута п. 2.3.

2.1.2. VMware ESX Server

Програмний продукт є комплексним рішенням для віртуалізації на підприємстві і виконує віртуалізацію на базі гіпервізора. Характерною ознакою даного продукту, яка відрізняє його від Microsoft Hyper-V і Xen, є відсутність потреби окремої керуючої ОС - її роль виконує спеціалізоване ядро Linux, що входить до складу продукту, що робить можливим його установку на вузли, де ОС відсутня в принципі. Отже, це вирішення відразу багатьох проблем, які напряму пов'язані з підбором і установкою керуючої ОС. [14]

Що стосується ОС VM, то VMware заявляє про підтримки будь-яких ОС - як 32, і 64-бітових, в залежності від наявності модифікацій в ядрі системи застосовується паравіртуалізація або повна віртуалізація. Підтримка паравіртуалізації пропонується для Linux, починаючи з ядра версії 2.6.21.

Також до складу даного продукту входить безліч інструментів і технологій для розгортання віртуальної інфраструктури в масштабах підприємств і

обчислювальних комплексів - інструменти для пакетного резервного копіювання, спільного використання мережевих сховищ даних і багато іншого.

2.1.3. Hyper-V

Компанія Microsoft розробила комплекс ПЗ до якого входить сам гіпервізор та система управління до нього. Він може працювати як окреме рішення, так і як частина ОС Microsoft Windows Server. [13]

Перша версія цього продукту була випущена в 2008 році. Він являє собою не що інше, як модуль ОС Windows Server 2008 або 2008 R2 (т.зв. «роль сервера»). Його установка можлива на сервери з 64-бітними процесорами, що працюють під керуванням 64-бітових версій Windows Server 2008 або 2008 R2 редакцій Standard, Enterprise і Datacenter.

Компоненти в архітектурі Hyper-V взаємодіють між собою користуючись таким поняттям як розділ. Розділ – це логічне поняття яке дозволяє розмежувати об'єкти системи. Розділи бувають двох типів кореневий і дочірні. Тільки один кореневий розділ може бути на одному фізичному сервері. У той час як кількість дочірніх розділів може доходити до 1024. Архітектура системи практично ідентична тій, що і у випадку з Xen: так само між ОС і устаткуванням вводиться шар гіпервізора, так само є окрема ОС, призначена для керування гіпервізором і має більше привілеїв у порівнянні з усіма іншими. У термінології Hyper-V екземпляри ОС називаються «розділами» (англ. partition), керуюча ОС і відповідне оточення іменується «батьківським розділом» (Англ. Parent partition), тоді як для позначення ВМ використовується термін «Дочірній розділ» (англ. Child partition).

ОС ВМ:

- будь-які версії і редакції Windows Server, починаючи з 2000 і закінчуючи 2008, включаючи підтримку 64-бітових систем;
- Red Hat Enterprise Linux 5.2, 5.3 і 5.4 - с підтримкою 64-бітних систем;
- SUSE Linux Enterprise Server версій 10 з пакетом оновлень 1 або 2, а також з підтримкою 64-бітних систем;
- клієнтські ОС Microsoft Windows, починаючи з Windows XP Service Pack2.

Призначені для домашнього використання (Home Basic і т.п.) версії Windows Vista і Windows 7 не підтримуються.

Також є дані про успішний запуск під Hyper-V систем Ubuntu Linux 6 і Fedora Core 8/9, хоча офіційно вони не підтримуються.

Для підтримки паравіртуалізації на VM з Linux Microsoft випустила комплект відповідних доповнень до ядра Linux, які офіційно були включені до складу вихідного коду цього ядра - однак, ці доповнення знаходяться під загрозою видалення з офіційного вихідного коду через недостатню їх підтримки з боку Microsoft.

Для підвищення продуктивності роботи з зовнішніми пристроями Microsoft постачає призначений для VM комплект доповнень, званий Enlightened I / O і працює з VM під керуванням Windows Server 2008, Windows 7 і SUSE Linux Enterprise Server.

2.1.4. KVM

Гіпервізор включений в ядро, за рахунок чого досягається дуже висока продуктивність віртуалізації. KVM є відкритим ПЗ, поширюється за ліцензією GPL v2 і складається з ядра, модуля взаємодії з процесором і компоненти користувальницького режиму. Для кращої продуктивності, процесор повинен підтримувати набір інструкцій для віртуалізації: Intel VT або AMD-V. KVM використовується як в корпоративних рішеннях (Redhat) так і для малих об'ємів віртуалізації. Вже кілька років ця технологія є основною для RHEL (Red Hat Enterprise Linux) замість XEN. [16]

Оскільки VM реалізована як Linux-процес, вона використовує стандартну модель безпеки Linux для ізоляції та управління ресурсами. З допомогою SELinux ядро Linux додає обов'язкові засоби контролю доступу, багаторівневі і різноманітні засоби захисту, а також управляє політикою безпеки. SELinux забезпечує сувору ізоляцію ресурсів і обмежує рухливість процесів, запущених в ядрі Linux.

KVM підтримує образи VM в розподілених файлових системах, таких як Global File System, так що вони можуть поділятися на кілька хостів або

узагальнюватися з використанням логічних томів. Підтримка тонкого налаштування образів дисків дозволяє оптимізувати використання ресурсів зберігання даних, виділяючи їх не відразу, а тільки тоді, коли цього вимагає ВМ. Власний формат дисків для KVM — QCOW2 забезпечує підтримку знімків поточного стану та забезпечує кілька рівнів таких знімків, а також стиснення і шифрування. KVM підтримує концепцію живої міграції, забезпечуючи можливість переміщення працюючих ВМ між фізичними вузлами без переривання обслуговування. Жива міграція прозора для користувачів. ВМ залишається включеною, мережеві з'єднання активними, і додатки користувача продовжують працювати, в той час як ВМ переміщається на новий фізичний сервер. Крім концепції живої міграції, KVM підтримує концепцію холодної міграції, дозволяючи зберігати і відновлювати її пізніше. KVM підтримує гібридну віртуалізацію, коли паравіртуалізовані встановлені драйвери у гостьовій ОС дозволяють ВМ використовувати оптимізований інтерфейс вводу/виводу, а не емулюють пристрій, забезпечуючи високу продуктивність введення/виводу для мережевих і блокових пристроїв.

2.2. Особливості міграції ВМ в хмарі

2.2.1. Огляд концепції. Міграція ВМ, тобто можливість переміщення ВМ цілком з одного обчислювальної конструкції без припинення її роботи на іншу, – є надзвичайно важливим завданням. Його вирішення дозволить покращити гнучкість використання ВМ і отримати відповідну економічну вигоду. Дійсно, міграція цілої ОС з усіма її додатками в комплексі дозволяє уникнути багатьох складнощів, які напряму пов'язані із проведенням міграції на рівні прикладних завдань. Крім того, міграція дозволяє отримати додаткову перевагу при використанні фізичних ресурсів – якщо є потреба у обслуговуванні апаратури, пов'язана з фізичним відмиканням вузлів, тоді міграція дозволяє перенести діючу систему з одного вузла на інший. Це надає повну свободу дій щодо вузлів з проблемами та жодним чином не обмежує призначену для користувача

функціональність: стан ВМ після міграції абсолютно ідентичний аналогічному стану до неї. [6]

Ще одним серйозним моментом є можливість поділу відповідальності користувачів ВМ і адміністратора обчислювальної системи: користувач не має надавати адміністратору які-небудь привілеї на доступ до ВМ для здійснення обслуговування, а адміністратор не повинен хвилюватися за ті процеси, які проходять всередині самої ВМ, так як міграція відбувається абсолютно незалежно від функціональності ВМ, на рівні гіпервізора, і є атомарною операцією.

Міграція ВМ є технічно складною багатоетапною процедурою, під час якої застосовуються складні алгоритми.

Зазвичай, в літературі визначено основними два варіанти міграції [6]:

- «швидка» (Quick migration) або «холодна» (Cold migration);
- «жива» міграція (Live migration).

Наведені типи різняться процесом, як між вузлами здійснюється перенесення сторінки пам'яті, що використовуються ВМ.

При «холодній» міграції вміст оперативної пам'яті ВМ зберігається на диску, потім ВМ переходить у стан «Сну», а користувачі не можуть працювати з нею. Потім стан машини повністю повідомляється на цільовий вузол, де цілим блоком проходить зчитування в оперативну пам'ять, а ВМ завантажується і робота продовжується - відповідно, всі процеси ВМ здійснюють перенесення на цільовий вузол і машина працює так, ніби міграції в принципі не було. Проблема тільки в тому, що на час проходження міграції користувачі не мають можливості працювати з ВМ.

«Жива» міграція є процесом значно складнішим. На відміну від «холодної» міграції ВМ не переходить у стан сну взагалі, а затримка, яка викликана зупинкою ВМ на початковому вузлі і повторним запуском її на цільовому вузлі, триває секунди. Досягнути цього вдається тому, що по мережі здійснюється перенесення всіх сторінок віртуальної пам'яті, що використовуються ВМ.

Відмінність між різними видами міграції: через істотно більш високу складність процесу перенесення сторінок пам'яті між вузлами процес «живої» міграції триває при інших рівних умовах більший час, ніж «холодної» - проте користувачі мають перевагу фактично нульового часу, коли ВМ недоступна.

Таким чином, вибір остаточної стратегії міграції повинен базуватися на тому, що в тій чи іншій ситуації важливіше - проміжок часу, який триватиме міграція, чи постійна доступність машини користувачам.

Необхідно відмітити, що виконання «живої» міграції є можливим лише тоді, коли ВМ розміщена на будь-якій файлової системі, що є спільною для вихідного і цільового вузла. Для отримання додаткової інформації про «живу» міграцію ВМ варто звернутися до статті розробників гіпервізора Xen [7]. Тут міститься повний опис технології «живої» міграції ВМ, які ним керуються.

2.2.2. Етапи процесу проведення міграції. На початковому рівні абстракції процес проведення міграції складається з наступних етапів:

- ініціалізація - щоб встановити з'єднання між фізичними обчислювачами і перенесення конфігурації ВМ, виділення ВМ пам'яті на цільовому вузлі. При "холодній" міграції виконується зупинка ВМ на вузлі її початкового встановлення;

- перенесення сторінок пам'яті - мабуть, найскладніший етап, який потребує застосування складних алгоритмів для його вирішення. Виконується в залежності від визначеного типу міграції - «швидкої» або «живої»;

- перенесення сховищ - якщо ВМ користуються єдиним сховищем даних, тоді контроль над певними файлами в такому сховищі на цей момент передається до цільового вузла. Якщо загальне сховище відсутнє, тоді виконується передача файлів по мережі на цільовий вузол;

- запуск перенесеної ВМ на цільовому вузлі. При «живій» міграції цьому процесу передують зупинка ВМ на вузлі її початкового розміщення.

2.2.3 Проведення міграції в гіпервізорах. Більшість основних сучасних гіпервізорів (Microsoft Hyper-V, VMWare ESX, Xen) використовують свій внутрішній засіб міграції ВМ. Можливості всіх реалізацій практично однакові і, коротко, всі необхідні умови її проведення можна звести до такого:

- процес міграції є можливим лише в межах однієї IP-підмережі (якщо Microsoft Hyper-V і VMWare ESX Server є одним кластером);

– процес міграції є можливим тільки тоді, коли наявне загальне мереже сховище даних, де розміщено віддалені вінчестери;

– моделі процесорів на обох вузлах (первинному і цільовому) мають бути якомога ближчі за типом, можливостями і наборами команд, в іншому випадку після виконання міграції комп'ютер може стати непрацездатним – творці гіпервізора не дають жодних гарантій щодо цього випадку.

Тут необхідно зробити застереження відносно першої умови. Для засобів Microsoft і VMWare є інші (не вбудовані) рішення і розробки, що дозволяють проводити перенесення між різними дата-центрами (тобто майданчиками розміщення серверів) і на достатні відстані, але за замовчуванням така можливість відсутня. Xen має можливість здійснювати "холодну" міграцію між різними підмережами, але це вимагає додаткового налаштування інтерфейсів мережі ВМ, яку переносять, так як внаслідок такої міграції без виконання додаткових налаштувань мережа для неї, найімовірніше, буде недоступною недоступна.

Що до другої умови, якщо відсутні додаткові заходи з перенесення сховищ, які забезпечуються окремими інструментами, наявність загального мережевого сховища даних є обов'язковим.

2.2.4 Про загальні мережеві сховища даних. В даний час існує декілька найбільш популярних концепцій і технологій організації загальних мережевих сховищ. Розглянемо найбільш відомі з них.

Мережа зберігання даних (SAN) - тип організації файлових сховищ, при якому створюється деяке архітектурне вирішення, що дозволяє серверам мережі працювати з підключеними до цієї мережі банкам даних як з локальними блоковими пристроями, наприклад, жорсткими дисками. Найбільш популярні технології, що працюють за цим принципом:

- iSCSI - транспорт SCSI за TCP / IP;
- ATAoE (ATA over Ethernet) - транспорт ATA в мережах на основі Ethernet;
- FCP (Fibre Channel Protocol) - транспорт SCSI в мережах на основі Fibre Channel;

Мережеве сховище (NAS) - при організації сховища за цим принципом доступ до даних забезпечується за допомогою мережевих файлових систем, таких як:

- NFS (Network File System) - мережева файлова система, широко застосовується на комп'ютерах з UNIX-подібними ОС;

- SMBFS / CIFS (Server Message Block File System з 1996 року – Common Internet File System) - стандартна мережева файлова система для комп'ютерів, що працюють під управлінням систем сімейства Microsoft Windows;

- AppleTalk - стек мережевих протоколів для комп'ютерів Apple Macintosh.

Основна відмінність між SAN і NAS зводиться до різниці між підходами до роботи з даними, застосовуваними при роботі з жорстким диском в одному випадку, і мережевої файлової системою - в іншому. У першому випадку до носія має доступ лише один клієнт, і робота з даними здійснюється за допомогою блочного читання і запису. У другому ж випадку мережевою файловою системою можуть користуватися багато клієнтів одночасно, а доступ до даних забезпечується на основі понять «файл» і «каталог».

2.3. Висновки до розділу

У цьому розділі описано поняття гіпервізора та досліджені його типи. Проаналізовано основні функції та можливості найбільш вживаних гіпервізорів (Xen, VMware ESX Server, Hyper-V, KVM). Досліджено особливості міграції ВМ в обчислювальній хмарі.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ XEN MIGR ДЛЯ УПРАВЛІННЯ ПРОЦЕСОМ МІГРАЦІЇ ВІРТУАЛЬНИХ МАШИН В ОБЧИСЛЮВАЛЬНІЙ ХМАРІ

3.1. Функціональні можливості та архітектура засобу

Спроектований засіб Xen Migr є програмним комплексом, що призначений для використання в обчислювальній хмарі. Його вузли функціонують під керуванням гіпервізора Xen. Основним завданням, який виконує програмний засіб, є надання адміністратору можливостей централізованого управління міграцією VM між вузлами хмари. Крім цього, інструментарій розробки включає в себе:

- засоби збору і відображення статистики використання обчислювальних ресурсів вузлами мережі і VM;
- складову управління файлами конфігурацій VM, яка працює разом із системою контролю версій Subversion (SVN);
- службу спостереження за станом вузлів хмари, що попереджає адміністратора мережі при падінні кількості вільних ресурсів на вузлах, нижче заданого порогу, або при відмові вузлів;
- засоби розгортання на вузлах хмари нових екземплярів VM, заснованих на шаблоні.

Архітектура програмного засобу Xen Migr наведена на рис.3.1. Визначено два види серверів - сервер-координатор, його роль виконує рівно один вузол, і довільну кількість серверів-агентів. Сервер-координатор пов'язаний з базою даних, в якій зберігається вся інформація, необхідна для функціонування системи, і SVN-репозиторієм, в якому зберігаються файли конфігурації віртуальних машин. Агенти є вузлами, що працюють під управлінням Xen, на яких безпосередньо розміщено віддалені машини - останні на рисунку зображені піктограмою у вигляді комп'ютера з підписом VM_i, де *i* - номер машини.



Рис. 3.1. Архітектура програмного засобу Xen Migr

Функції агентів полягають у наступному:

- періодична відправка координатору даних про завантаженість ресурсів вузла;
- надання відповідним VM середовища виконання;
- прийом від координатора запитів на розгортання нової VM, заснованої на шаблоні;
- прийом від координатора запитів на міграцію VM на інший вузол.

VM, як і агенти, надають координатору дані про завантаженість ресурсів, але в міграції як вузли розміщення VM, звичайно, участі не беруть .

Функції координатора:

- обробка та агрегування даних про завантаженість ресурсів вузлів мережі;
- надання адміністратору веб-інтерфейсу для управління засобом;
- спостереження за станом вузлів на основі переданої ними інформації та видача попереджень при падінні кількості ресурсів на вузлі нижче заданого порогу або відсутності активності з боку вузла протягом тривалого часу;
- пошук варіантів вузла призначення при необхідності міграції VM.

Системні служби агентів призначені для запуску в середовищі ОС Linux, оскільки включають в себе специфічний для цієї системи код - зазвичай саме ця система використовується в якості керуючої на вузлах, що працюють під управлінням Xen. Служби же координатора можуть бути використані в будь-якому середовищі.

На основі наведеного вище можна виявити такі процеси, що відбуваються в хмарі, ініційовані і керовані системою Xen Migr:

- збір і обробка статистики використання ресурсів обчислювальних вузлів і VM;
- спостереження за станом вузлів і при необхідності видача попереджень;
- розгортання нових екземплярів VM на основі шаблону;
- міграція VM.

Розглянемо деталі реалізації програмного засобу докладніше, починаючи з огляду його програмної архітектури.

3.2. Програмна архітектура засобу

Як видно з п. 3.1, компоненти Xen Migr можна умовно розділити на дві групи - системні служби координатора і агентів і компоненти для користувача інтерфейсу.

Системні служби розроблені з використанням мови програмування Java та декількох бібліотек класів мови, а веб-інтерфейс - засобами мови програмування PHP і її бібліотек.

Програмна структура системи з точки зору служб на базовому рівні абстракції, що включає в себе пакети класів, представлена на рис. 3.2:

Основним метапакетом засобу є пакет `xenmigrmotion` (повне ім'я - `org.xenmigrmotion`), що містить в своєму складі більше вузькоспеціалізовані пакети.

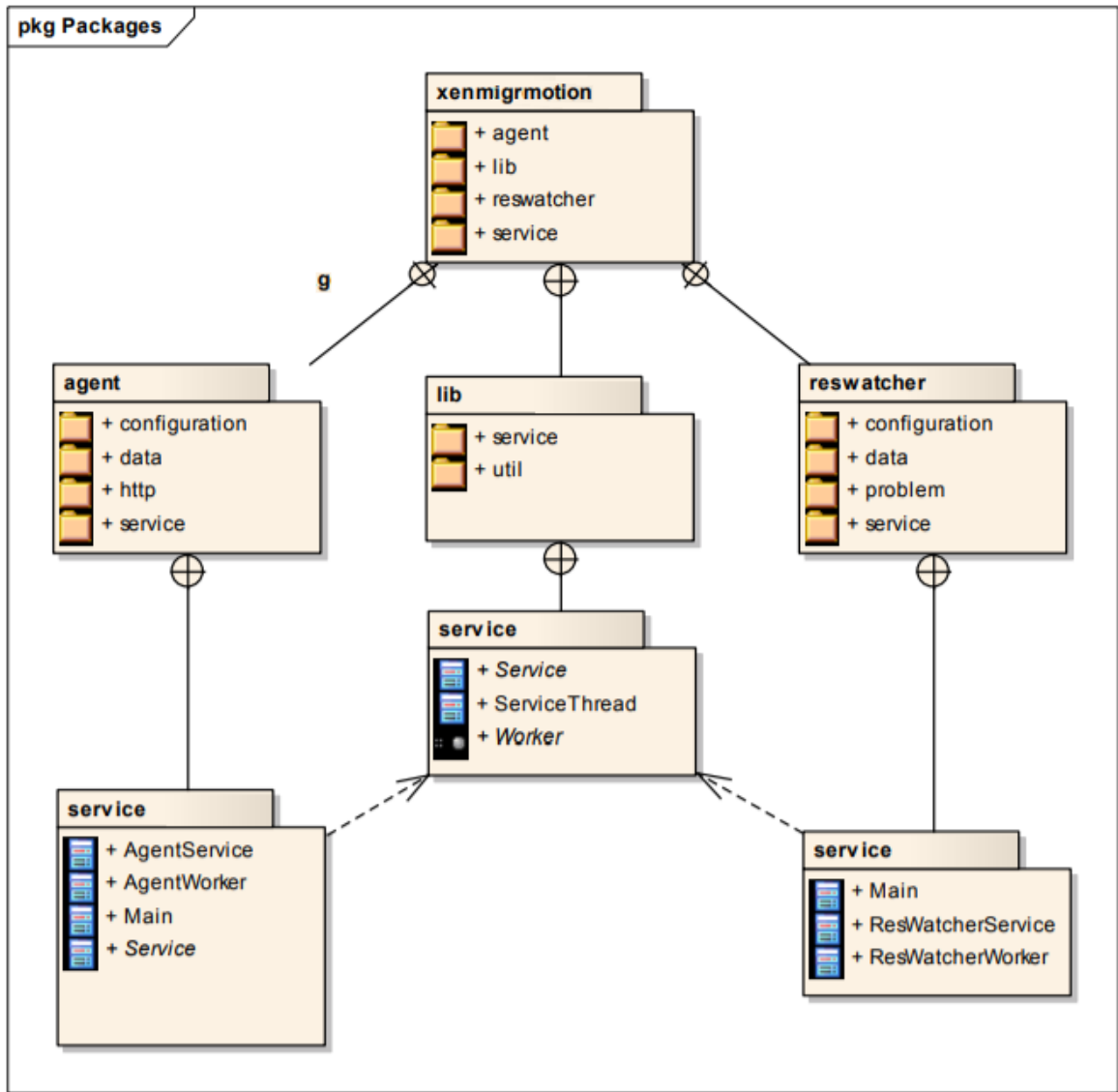


Рис. 3.2. Основні пакети класів служб засобу Xen Migr та відношення між ними

Пакет `xenmigrmotion.lib` включає в себе пакети `service` та `util`, в яких містяться відповідно класи для забезпечення можливості запуску компонентів, таких як служб UNIX (демонів) або служб Windows (останнє може бути актуально при кросплатформовому використанні служб координатора) і класи загального призначення - наприклад, що забезпечують шифрування рядків за алгоритмом MD5.

Пакет `xenmigrmotion.agent`, як видно з назви, включає в себе класи, які забезпечують функціональність служб засобу на стороні агента, зокрема:

- служба періодичної відправки координатору даних про завантаженість ресурсів вузла (включає в себе класи, які виконують отримання даних про поточну завантаженість ресурсів, а також формування і відправку координатору HTTP-запитів, що містять ці дані);

- класи для роботи з XML-файлом конфігурації агента;
- класи для забезпечення зворотного зв'язку з координатором за допомогою вмонтованого HTTP-сервера (для ситуацій, коли на вузлі потрібно створити на основі шаблону нову VM або зробити міграцію вже існуючої VM).

Пакет `xenmigration.reswatcher` містить в своєму складі службу, призначену для виконання на стороні координатора і спостереження за станом вузлів на основі переданих ними даних. Його складові такі:

- служба періодичної перевірки наявності проблем на вузлах;
- класи для роботи із XML-файлом конфігурації служби.

3.3. Організація роботи системних служб

На принципах роботи компонентів, організованих за принципом системної служби, необхідно зупинитися трохи докладніше, оскільки на цих принципах ґрунтується робота декількох важливих частин сервісу, і вони є концептуально загальними для цих частин.

Для забезпечення роботи цих компонентів використовується пакет `Apache Commons Daemon`. Його призначення полягає у тому, щоб забезпечувати Java-додатками середовище виконання і оточення, аналогічні до тих, що зазвичай мають системні служби в будь якій ОС - таким чином відпадає необхідність писати системний код більш низькорівневою мовою програмування. Даний пакет є кросплатформним, і тому може використовуватися для розробки системних служб як для UNIX-подібних ОС, так і для систем сімейства Windows . Цей пакет включає в себе, по-перше, програмний інтерфейс системної служби, написаний на мові Java, а по-друге - виконуваний файл, який безпосередньо і виконує запуск Java-коду в оточенні системної служби. Власне кажучи, за допомогою використання цього файлу як контейнера для запуску Java-коду в якості системної служби програміст і звільняється від необхідності написання системного коду. Природно, до складу пакету включаються версії цього файлу, призначені для запуску в середовищі різних ОС - саме таким чином і забезпечується кросплатформеність цього рішення.

У дипломній роботі цей пакет Apache Commons Daemon використаний в такий спосіб:

- створено клас `ServiceThread`, який повинен інкапсулювати основний робочий цикл системної служби в рамках потоку та включає в себе логіку для запуску і зупинки цього циклу;

- створено інтерфейс `Worker`, реалізації якого покликані інкапсулювати власне функціональність служби - на кожному кроці основного робочого циклу її виконання викликається метод `work` відповідної реалізації цього інтерфейсу.

- створено абстрактний клас `Service`, що включає в себе логіку управління потоком класу `ServiceThread`, а також абстрактний метод `getWorker()` – всі нащадки цього класу, перевантажуючи цей метод, будуть таким чином надавати вищезазначеного потоку об'єкт деякого класу, реалізовує інтерфейс `Worker` відповідно з функціональним призначенням розроблюваної служби.

Структурні відношення між вищезгаданими класами і інтерфейсами представлені на рис. 3.3.

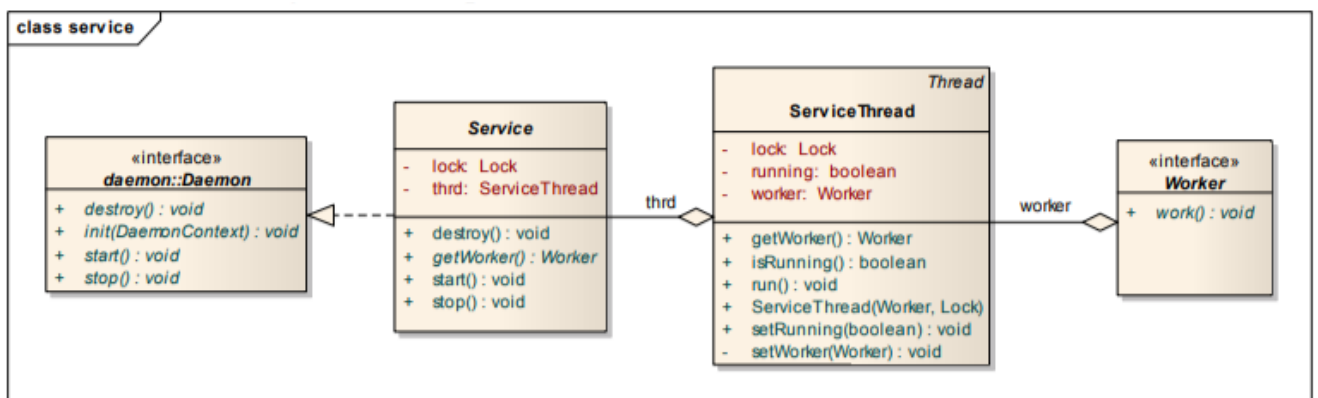


Рис. 3.3. Діаграма класів, що забезпечують роботу системних служб

Наступна діаграма послідовностей (рис. 3.4) демонструє динамічні відношення між об'єктами цих класів і інтерфейсів - в загальному випадку, без деталей конкретної реалізації.

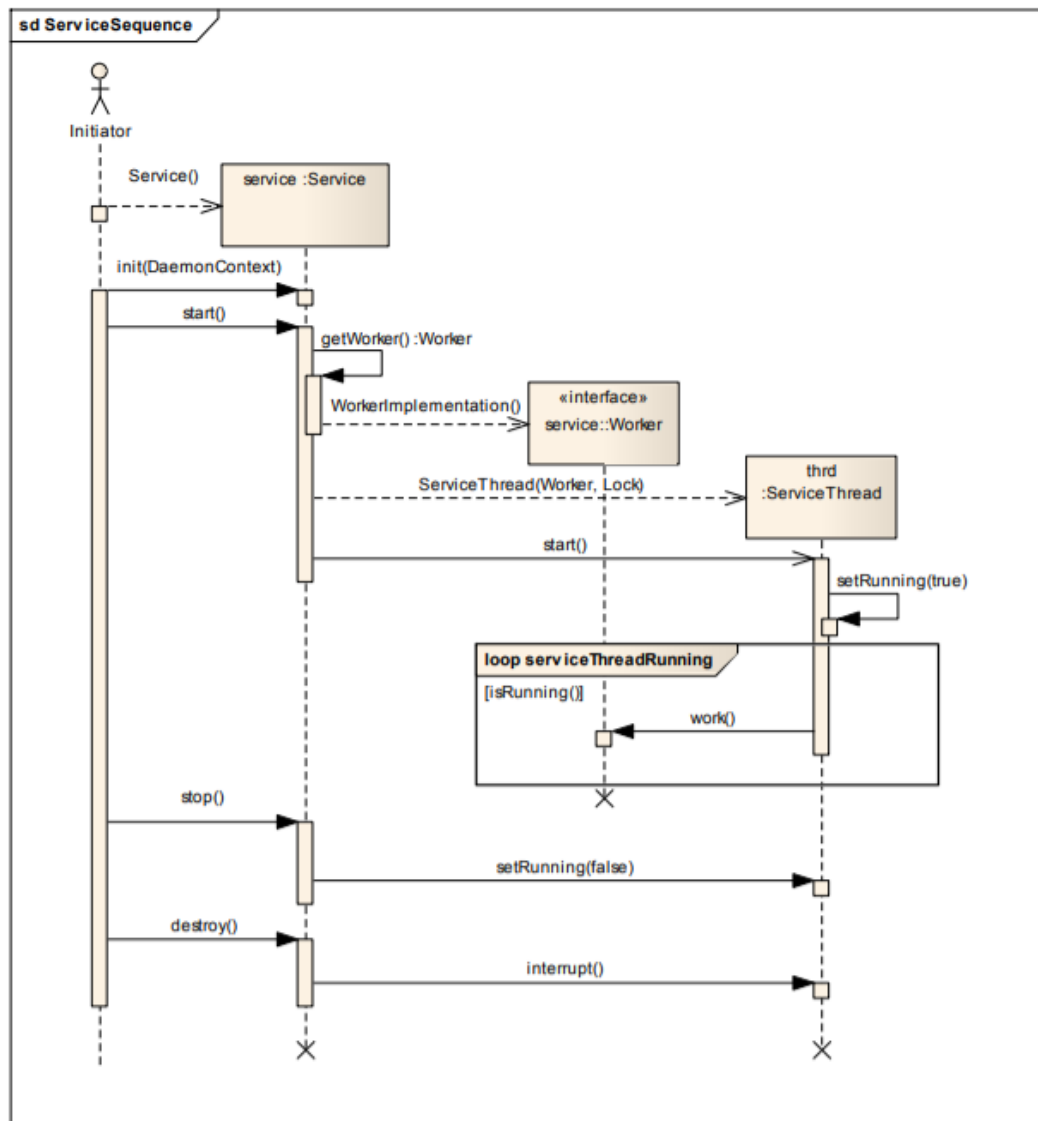


Рис. 3.4. Діаграма послідовностей об'єктів, що забезпечують роботу системних служб, базовий випадок

3.4. Основні процеси в хмарі

На основі сказаного вище про процеси у хмарі і програмної архітектури засобу можна розглянути ці процеси більш детально. Опис процесів включатиме в себе опис логіки роботи відповідних частин програмного засобу, технічні деталі їх реалізації, включаючи опис використаних у розробці сторонніх програмних бібліотек, опис користувацького інтерфейсу, а також структуру і відносини відповідних класів і сутностей в БД.

3.4.1. Збір статистики використання ресурсів вузлами хмари

Цей процес дозволяє виконувати наступні завдання:

- надання адміністратору можливості спостереження за динамікою споживання ресурсів на вузлах, яка змінюється з плином часу;
- пошук варіантів цільового вузла при необхідності міграції ВМ відповідно до потреб останньої;
- спостереження за станом вузлів мережі.

На рис. 3.5 подано діаграму основних класів, які використовуються даним процесом.

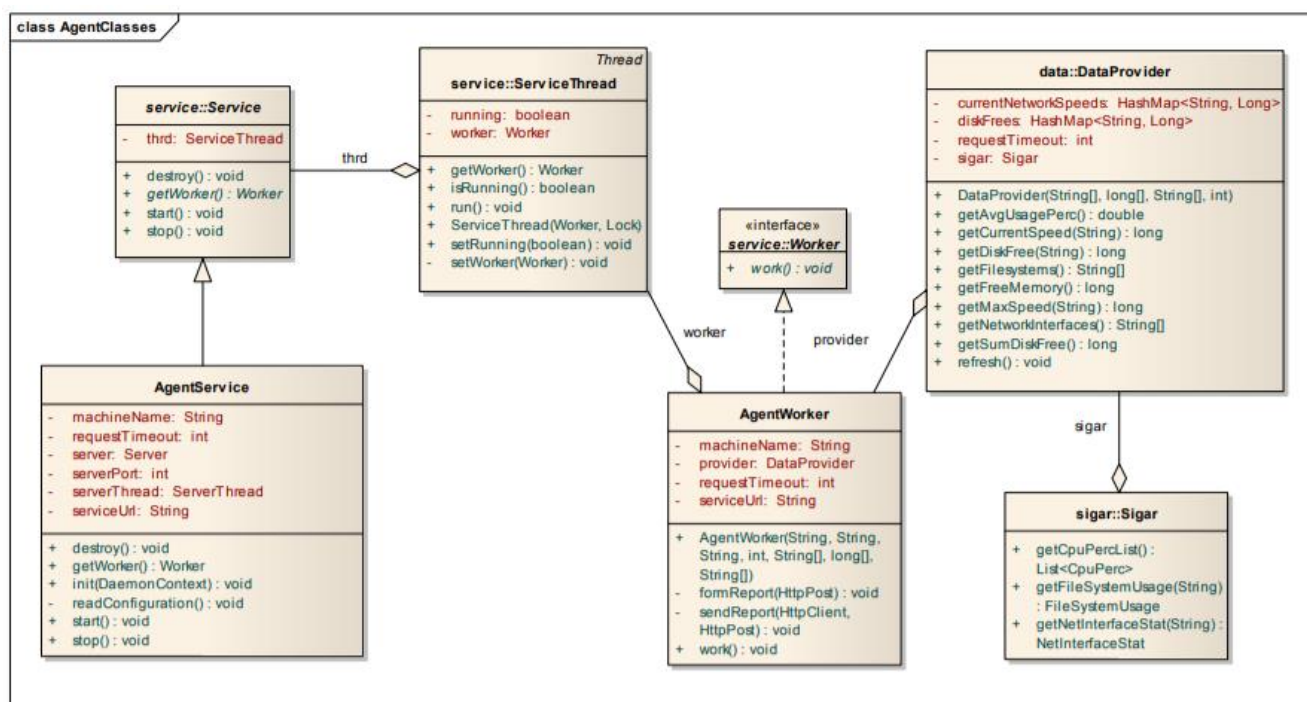


Рис. 3.5. Діаграма основних класів, які використовуються службами агента

Найбільш важливими класами на цій діаграмі є класи `AgentService`, `AgentWorker` і `DataProvider`. Перші два, як видно з діаграми, ґрунтуються на вже згадуваних в п. 3.3 класі `Service` і інтерфейсі `Worker` відповідно, і їх функціональне призначення впливає з уже описаного призначення їхніх предків, яке вони лише конкретизують в контексті розроблюваної служби: `AgentService` інкапсулює основну службу агента, що відповідає за збір статистики, і управління вбудованим HTTP-сервером (про нього і його призначення мова піде в іншому розділі), а `AgentWorker` на кожній ітерації основного робочого циклу служби збирає дані про використанні ресурсів і відправляє їх координатору за допомогою

HTTP POST-запиту. Для збору даних про завантаженість ресурсів використовується клас `DataProvider`, який інкапсулює роботу з призначеною для цих цілей бібліотекою `Hyperic SIGAR`. Відносини між класами `AgentService` і `AgentWorker`, а також класами, описаними в п. 3.3, відповідають описаному раніше механізму роботи класів, що забезпечують функціонування системних служб.

Для здійснення збору даних про завантаженість вузла і відправки цих даних службам агента необхідні деякі налаштування, які винесені в файл конфігурації служби. Цей файл є документом формату XML, робота з яким здійснюється за допомогою бібліотеки `Apache Commons Configuration`, і містить такі дані:

- ім'я вузла і його пароль для відправки даних координатору;
- інтервал відправки даних про завантаженість вузла координатору;
- адреса (URL) служби координатора, за якою будуть відправлятися дані чергових вимірювань завантаженості ресурсів вузла;
- список файлових систем, щодо яких необхідно здійснювати збирання статистики вільного простору;
- список мережевих інтерфейсів, щодо яких необхідно здійснювати збір статистики їх завантаженості з опціональним зазначенням їх максимальних швидкостей.

На етапі ініціалізації об'єкта класу `DataProvider` на основі файлу конфігурації служби проводиться ще кілька важливих операцій. По-перше, аналізується список файлових систем, щодо яких необхідно збирати статистику, або, при відсутності в файлі конфігурації відповідного списку, вважається, що аналізувати необхідно всі файлові системи; в підсумкової вибірці беруть участь лише локальні жорсткі диски і мережеві файлових систем. По-друге, аналізується список мережевих інтерфейсів, щодо яких необхідно збирати статистику, або при відсутності в файлі конфігурації відповідного списку вважається, що аналізувати необхідно всі мережеві інтерфейси. Якщо список інтерфейсів сформований динамічно, або для будь-якого інтерфейсу в файлі конфігурації не вказано вручну максимальна пропускна здатність, то об'єкт класу `DataProvider` за допомогою системних викликів, специфічних для UNIX, визначає максимальну швидкість цих інтерфейсів.

Також в цьому файлі містяться параметри конфігурації вбудованого HTTP-сервера даної служби, про них мова піде вподальшому.

Варто зупинитися докладніше на процедурі збору і відправки даних про завантаженість ресурсів, яка є основним завданням даної служби. Як уже було відзначено вище, для збору даних про поточну завантаженість ресурсів вузла використовується бібліотека Hyperic SIGAR. Це кросплатформена бібліотека, містить засоби для вимірювання поточної завантаженості процесора, вільного простору на жорстких дисках, кількості вільної оперативної пам'яті і спостереження за мережевою активністю, а також багато чого, що розповсюджується за ліцензією GNU (GPL) та має інтерфейси для роботи з нею з середовищ Java, Microsoft .NET, Perl і інших.

Етапи цієї процедури такі:

- збір даних про завантаженість ресурсів вузла;
- вимір завантаженості процесорів вузла;
- вимір кількості вільної оперативної пам'яті на вузлі;
- вимір кількості вільного простору на файлових системах вузла;
- вимір завантаженості мережевих інтерфейсів вузла;
- формування звіту;
- відправка звіту.

Збір статистики про використання ресурсів здійснюється за такими періодам:

- за останню добу з кроком в 5 хвилин;
- за останній тиждень з кроком в 30 хвилин;
- протягом останнього місяця з кроком в 1 годину;
- за останній рік з кроком в 1 добу

Для вимірювання завантаженості процесорів вузла вимірюється завантаженість кожного з них окремо як відсоток часу, що витрачається на виконання різних операцій, і знаходиться середнє арифметичне цих значень, прийняте в якості остаточного коефіцієнта завантаженості процесорів вузла. Вимірювання кількості вільної оперативної пам'яті є атомарною операцією бібліотеки SIGAR, для отримання цих даних достатньо виклику лише одного методу. Вільний дисковий простір обчислюється атомарному для кожної файлової

системи зі списку - також викликом єдиного методу SIGAR. Для вимірювання використовуваної смуги пропускання на мережевому інтерфейсі за допомогою SIGAR добуваються значення відправлених та отриманих байт для даного інтерфейсу, далі обчислюються різниці цих показників і відповідних даних попереднього вимірювання (ці дані зберігаються всередині об'єкта `DataProvider`). Отримані значення підсумовуються - так ми отримуємо сумарну кількість байт, переданих цим інтерфейсом протягом останнього інтервалу робочого циклу служби. Далі ця сума ділиться на довжину інтервалу - таким чином ми отримуємо поточну швидкість передачі даних на інтерфейсі, яка представляє собою усереднене значення швидкості за період останнього інтервалу - наприклад, останні 5 секунд. Потім дані останнього виміру відправленого та отриманого кількості байт для даного інтерфейсу зберігаються. Дана операція здійснюється для всіх елементів списку мережевих інтерфейсів, що підлягають збору статистики.

Діаграма послідовностей, представлена на рис. 3.6, наочно зображує процес вимірювання завантаженості ресурсів вузла.

На етапі відправки запиту сформований HTTP POST-запит відправляється за адресою служби координатора, взятому з файлу конфігурації, потім робочий потік служби (об'єкт класу `ServiceThread`) «засинає» на період інтервалу, також вказаний в файлі конфігурації агента - потім настає нова ітерація робочого циклу служби, і процес триває. Власне запит інкапсулюється об'єктом класу `HttpPost` з багатофункціональної бібліотеки класів `Apache Commons HTTP Client` і відправляється координатору засобами цієї ж бібліотеки.

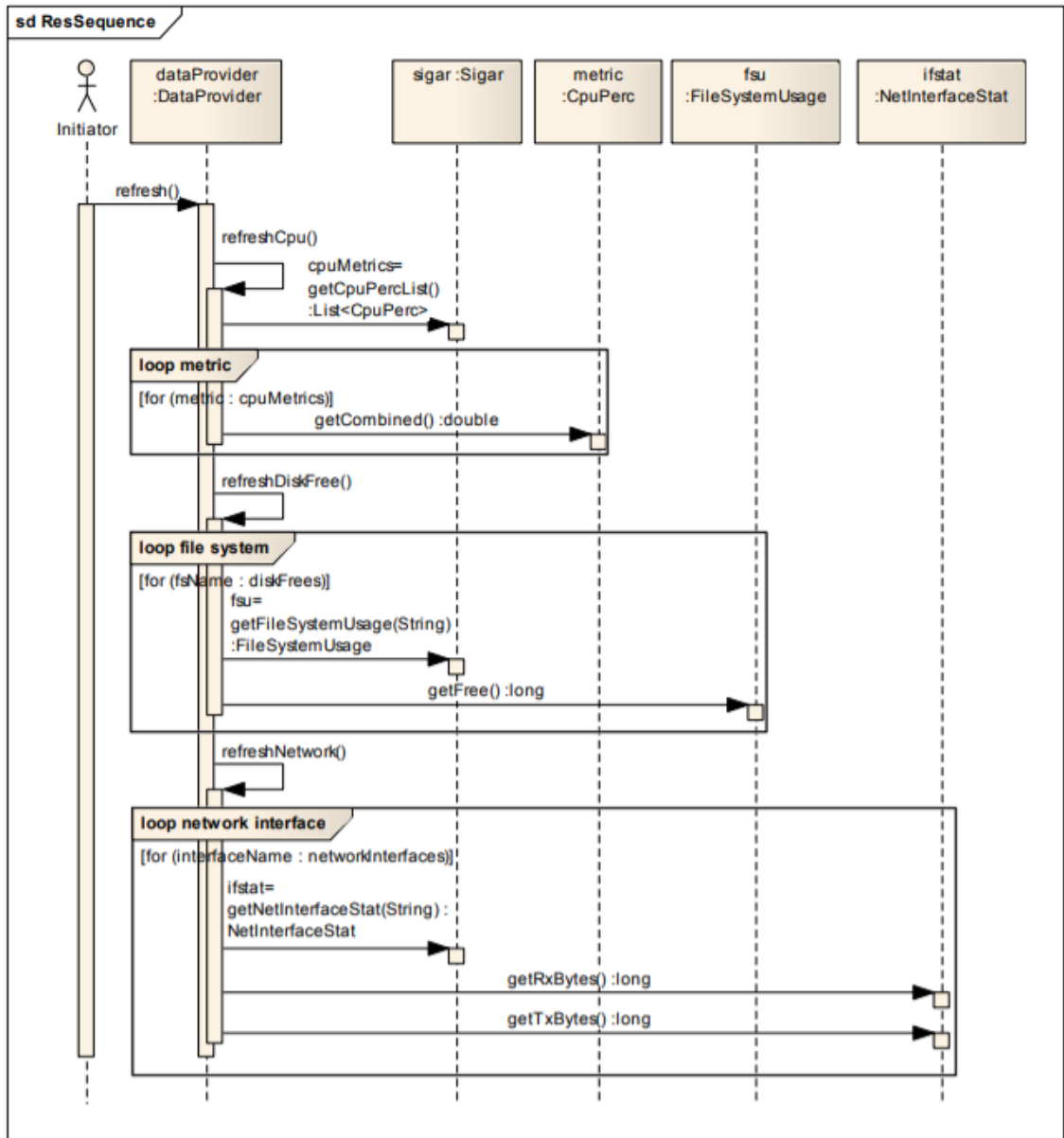


Рис. 3.6. Діаграма послідовностей об'єктів, які використовуються для вимірювання поточної завантаженості ресурсів вузла (ітерація основного циклу служби)

Обробка запитів, що надходять із даними про завантаженість вузлів відбувається на координаторі за допомогою сценарію на мові RNR. Його роботу можна розділити на наступні етапи:

- перевірка коректності переданих даних і авторизація запиту;
- збереження містяться в запиті даних;
- оновлення статистики використання ресурсів.

Дії на етапі верифікації запиту;

- по-перше, проводиться перевірка наявності в тілі запиту всіх обов'язкових полів - якщо хоча б одне з них не виявлено, то обробка запиту припиняється;

– по-друге, проводиться пошук в БД вузла з ім'ям, зазначеним в полі запиту `machineName` - якщо в результаті цього пошуку не вдається знайти зареєстрований вузол з таким ім'ям, або збережена в базі IP-адреса не збігається з адресою, з якої був відправлений запит, то обробка цього запиту також припиняється;

– по-третє, за допомогою ключа запиту здійснюється його авторизація - з БД добувається MD5-хеш пароля, що відповідає цьому вузлу, далі на основі імені вузла і параметра запиту `requestTime` за описаним генерується ключ. Якщо отриманий ключ збігається з переданим в поле запиту `requestKey` значенням, то запит вважається авторизованим і його обробка продовжується, інакше сценарій завершує виконання.

Для верифікації запиту використовується таблиця БД `host`, що містить дані про всі вузли хмари, які використовуються системою, і всі VM, які виконуються на цих вузлах. З точки зору служб статистики використання ресурсів і контролю за станом вузлів немає різниці між VM і реальним вузлом - таким чином, VM можуть відправляти координатору дані про завантаженості ресурсів нарівні з реальними вузлами, використовуючи ті ж класи і інтерфейси та набір таблиць БД. Реальний вузол відрізняє від віртуальної машини значення «фальш» в полі `vm` рядку таблиці `host` БД - VM мають в цьому полі значення «істина».

3.4.2. Спостереження за станом вузлів хмари

Цей процес дозволяє адміністратору здійснювати спостереження за працездатністю вузлів обчислювального комплексу, виявляючи відмови і надмірно завантажені вузли, а також таким же чином стежити за станом VM які виконуються в хмарі.

Наведена на рис. 3.7 діаграма класів демонструє основні класи, які використовує даний процес, а також структурні відносини між ними.

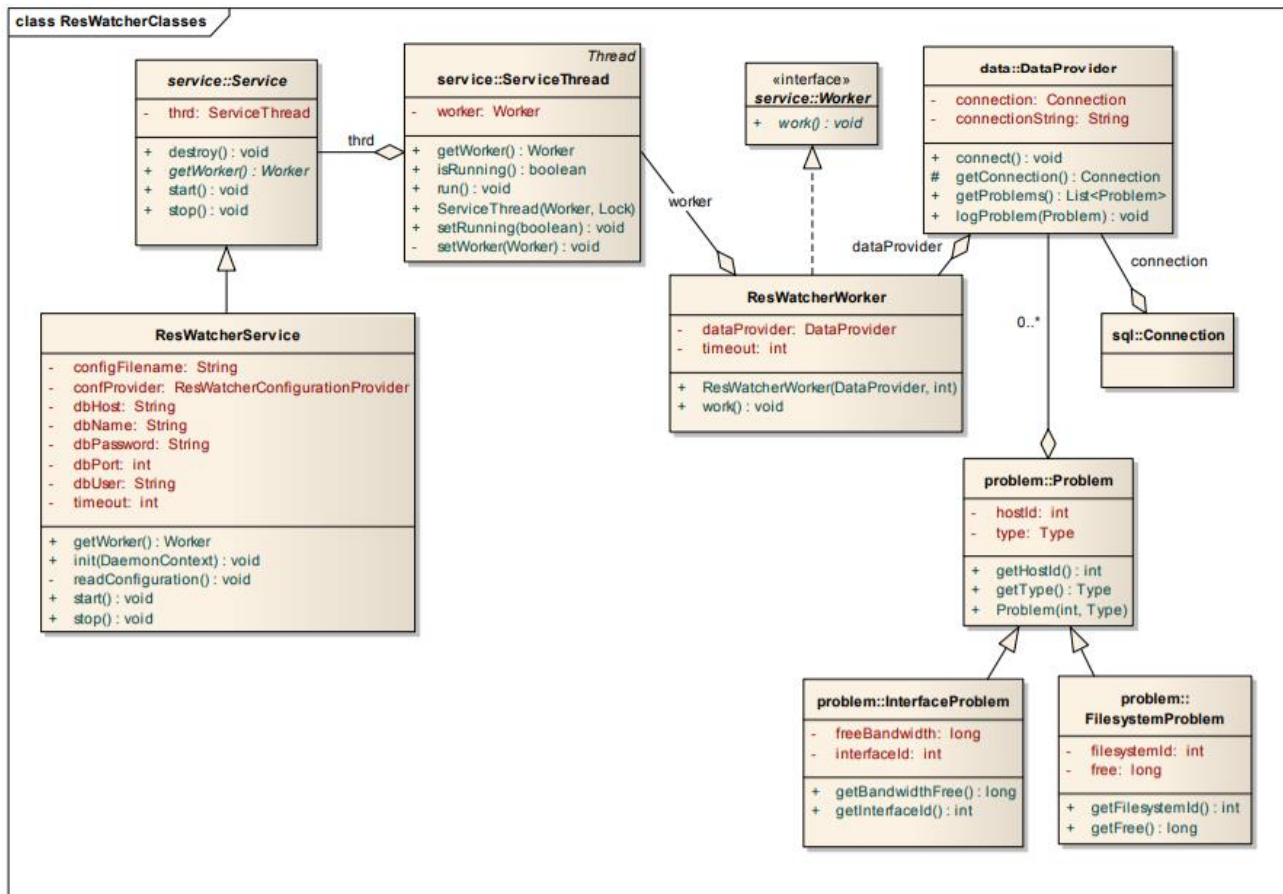


Рис. 3.7. Основні класи, які використовує служба спостереження за станом вузлів

Класи ResWatcherService і ServiceThread є відповідно нащадком класу Service і реалізацією інтерфейсу Worker, описаних в п. 3.3, і за своїм призначенням концептуально близькі до класів AgentService і AgentWorker, описаних в п. 3.4.1, будучи, по суті, інкапсуляцією основного робочого циклу даної служби. Основні функції даного процесу реалізовані за допомогою класів DataProvider і Problem. Перший інкапсулює роботу з БД MySQL, здійснюючи власне спостереження за станом вузлів, а другий являє собою контейнер для окремого попередження, що містить ідентифікатор проблемного вузла і тип повідомлення про помилку, представлений у вигляді числового коду. Типи можливих помилок такі:

- відсутність запитів з оновленнями даних про завантаженість ресурсів вузла упродовж часу, рівного двом інтервалам відправки даних цим вузлом - можливий загальний відмову вузла внаслідок перевантаження (код 0);
- перевантаженість процесорів вузла (оцінюється за середнім коефіцієнтом) (код 1);
- перевантаженість оперативної пам'яті вузла (код 2);

- падіння максимальної кількості вільного дискового простору серед файлових систем вузла нижче заданого порогу (код 3);
- падіння сумарної вільної пропускнуої спроможності мережевих інтерфейсів вузла нижче заданого порогу (код 4);
- падіння кількості вільного дискового простору на одній з файлових систем вузла нижче заданого порогу (код 5);
- падіння вільної пропускнуої спроможності на одному з мережевих інтерфейсів вузла нижче заданого порогу (код 6).

Оскільки інкапсуляція повідомлень про помилки останніх двох типів вимагає додаткової інформації, а саме ідентифікаторів проблемних файлових систем і мережевих інтерфейсів відповідно, для цієї мети було введено два класи-нащадка класу `Problem` - відповідно класи `FilesystemProblem` і `InterfaceProblem`.

Оскільки дана служба виконується на координаторі та працює безпосередньо з БД, її файл конфігурації містить дані, необхідні для виконання цього завдання, а саме: ім'я або IP-адресу сервера MySQL, порт доступу до цього сервера, ім'я цільової БД, а також ім'я користувача та пароль для доступу до неї. Також в файлі конфігурації є параметр таймаута робочого циклу служби.

Процедура перевірки стану вузлів виконується циклічно з таймаутом, значення якого зчитується з файлу конфігурації. Алгоритм цієї процедури досить простий: з уявлень (англ. `views`) БД, що містять дані про перевантажені ресурси вузла, добуваються відповідні записи, і на основі цих даних проводиться запис повідомлень про помилку в таблицю БД `problem_log` - зберігаються ідентифікатор вузла, тип помилки і час запису. Якщо в таблиці `problem_log` вже є запис про помилку деякого типу для заданого вузла, то вставка нового запису про помилку не відбувається, натомість лічильник повторень помилки збільшується на одиницю, а час виникнення помилки змінюється на поточний. Після цього робочий цикл служби переходить до наступної ітерації.

Необхідно згадати про те, як оцінюється той чи інший перевантажений ресурс вузла - інакше кажучи, звідки беруться значення для порівняння з останніми даними про завантаженість ресурсів вузла або ВМ. Ці значення зчитуються з таблиці БД `host_thresholds` - між цією таблицею і таблицею `host` існує ID-залежність, у якій роль сильної сутності виконує,

природно, остання таблиця. У таблиці `host_thresholds` кожне поле відповідає одному з типів ресурсів, і значення цього поля вказує критичний поріг завантаженості відповідного ресурсу на відповідному вузлі. Якщо в таблиці `host_thresholds` немає запису, що відповідає таблиці `host`, то вважається, що обмежень на використання вузлом ресурсів немає. Якщо ж такий запис існує, але у одному або декількох його полях, відповідних типів ресурсів, зберігається значення `NULL` або `0`, то вважається, що використання відповідного ресурсу не обмежене. Крім зведеної таблиці з обмеженнями, якої є `host_thresholds`, для завдання порогів використання конкретних пристроїв - файлових систем і мережних інтерфейсів - використовуються відповідно поля `free_space_threshold` таблиці `filesystem` і `free_bandwidth_threshold` таблиці `network_interface`. Значення полів цих таблиць порівнюються з відповідними даними статистики використання ресурсів впродовж останніх 5 хвилин - це зроблено для того, щоб запобігти появі повідомлень про помилку при виникненні короткочасних стрибків завантаження.

Для того, щоб максимально спростити програмний код служби і зробити атомарну транзакцію до БД, необхідну для власне проведення перевірки стану вузлів, була спроектована кілька подань бази даних (англ. `views`), всередині яких проводиться об'єднання (англ. `join`) відповідних таблиць і відповідні перевірки. Результати всіх перевірок легко витягуються з уявлення `view_host_health`, кожне поле якого містить значення «істина», «брехня» або `NULL`. Якщо в деякому полі будь-якого рядка цього подання встановлено значення «істина», то це означає, що відповідний ресурс перевантажений. Будь-яке інше значення означає, що перевантаження ресурсу немає, або його використання необмежено.

На рис. 3.8 наведено відношення між основними таблицями і сутностями.

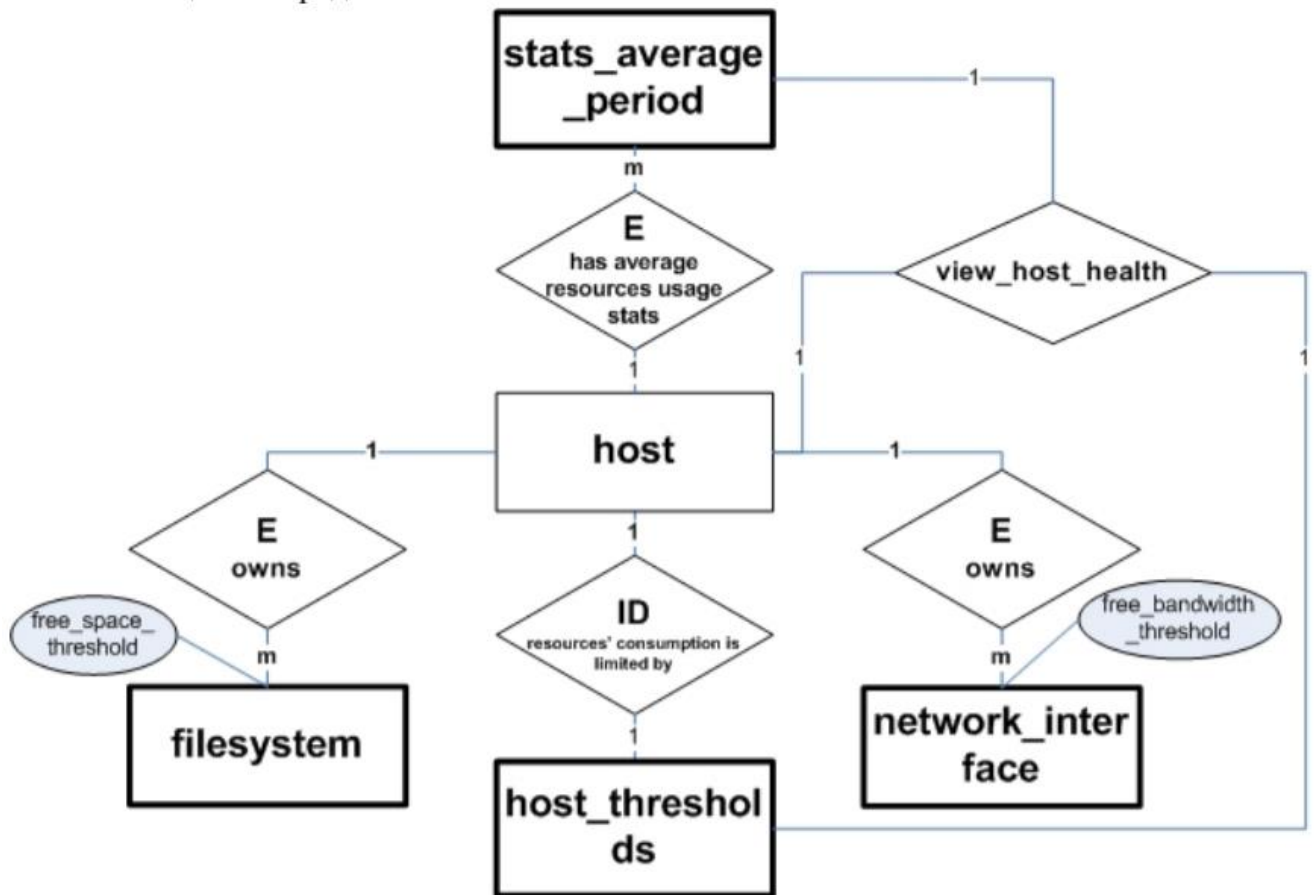


Рис. 3.8. ER-діаграма основних сутностей БД, що використовуються процесом спостереження за станом вузлів хмари

3.4.3. Розгортання нових екземплярів ВС на основі шаблону

За допомогою засобів цього процесу адміністратор отримує можливість автоматичного розгортання (тобто створення і запуску) на деякому вузлі хмари VM, заснованої на деякому шаблоні. Наявність в системі такої функції дозволяє підвищити гнучкість використання обчислювальних ресурсів і ширше охопити весь спектр операцій над VM.

Основну частину операцій в ході даного процесу виконує модифікований сценарій `xmclone.sh`, що поставляється разом з Xen [8]. В цей сценарій було внесено модифікації, що дозволяють виробляти копіювання VM в ситуаціях, коли вихідна і цільова папки з файлами конфігурації VM, а також вихідна і цільова папки з образами віртуальних жорстких дисків VM різні. Копіювання файла конфігурації і при необхідності - файлів віртуальних жорстких дисків шаблону проводиться по мережі: на агента монтуються відповідні мережеві каталоги

протоколу NFS. Шляхи до цих файлів щодо локального дерева каталогів агента задаються у файлі конфігурації служби. Алгоритм наступний:

- адміністратор вибирає цільовий вузол розміщення нової VM;
- вибираються IP- і MAC-адреси для мережевого інтерфейсу нової VM – це необхідно зробити для того, щоб ця машина відразу ж після свого запуску на цільовому вузлі мала доступом до мережі на основі унікальних мережеских реквізитів. Вибір проводиться вручну або автоматично з фіксованих пулів адрес;
- координатор відправляє агенту HTTP-запит, що інкапсулює необхідні параметри процедури розгортання;
- на агента з відповідними параметрами запускається сценарій копіювання шаблону VM на його локальну файлову систему:
 - копіюються файли або розділи віртуальних жорстких дисків шаблону VM;
 - копіюється файл конфігурації VM;
 - редагується файл конфігурації VM;
 - локально монтується основний віртуальний жорсткий диск VM;
 - вносяться зміни до відповідних системних файлів на диску VM відповідно до параметрів, переданими в HTTP-запиті;
 - координатору повідомляється про те, що новий екземпляр VM успішно створений;
 - нова VM запускається.

Як видно із вищенаведеного списку основна частина роботи, яка виконується в рамках даного процесу, відбувається на кроці 4 алгоритму. Виконується ця робота, як уже було сказано вище, модифікованим сценарієм `xmclone.sh`.

На рис. 3.9 відношення між відповідними класами представлені у вигляді діаграми.

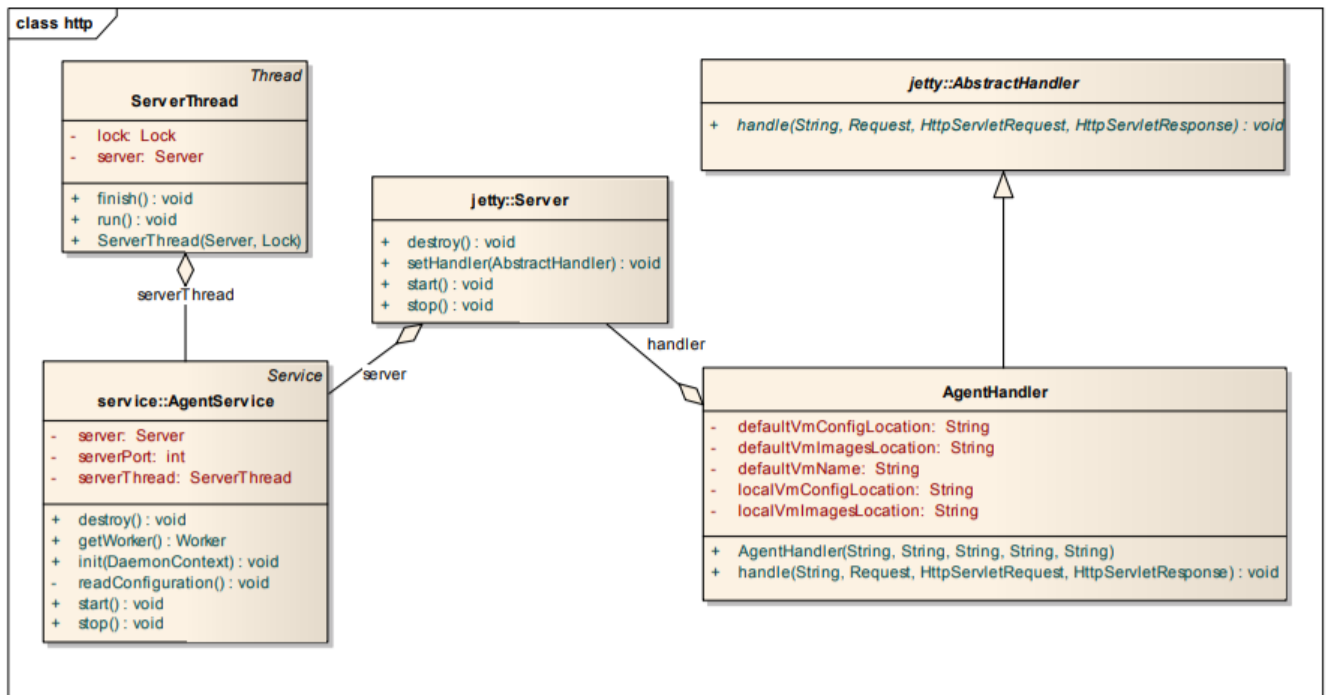


Рис. 3.9. Діаграма класів, що беруть участь в обробці HTTP-запитів

Даний процес виконується на дещо нижчому рівні, ніж інші процеси, описані раніше. Це пов'язано з тим, що для його здійснення необхідно використання команд гіпервізора і декількох системних команд, майже всі з яких викликаються сценарієм `xmclone.sh`. Тому з класів мови Java в даному процесі була використана лише бібліотека Jetty, що дозволяє вбудовувати в вже існуючі програми HTTP-сервер, що працює в окремому потоці. Це було зроблено для того, щоб забезпечити можливість відправки координатором агенту деяких повідомлень - таких, наприклад, як команда міграції або розгортання ВМ. Для вбудовування Jetty в архітектуру системи були додані класи потоку, всередині якого виконується сервер (`ServerThread`), а також обробника запитів (`AgentHandler`, нащадок `AbstractHandler`). Об'єкт класу `ServerThread` є атрибутом описаного в п. 3.4.1.

Крім згаданих шляхів до файлів шаблону ВМ, у файлі конфігурації агента також містяться налаштування, необхідні для роботи вбудованого HTTP-сервера, такі, наприклад, як номер порту, на якому він буде відповісти на вхідне з'єднання.

Існують ряд обмежень, які необхідно враховувати:

- цей процес пов'язаний з переносом по мережі великої кількості даних - обсяг віртуального жорсткого диска з сучасною ОС цілком може досягати декількох гігабайтів, і, відповідно, може займати значну кількість часу. Втім, є

можливість скопіювати файли шаблону VM локально і задати у файлі конфігурації агента відповідні шляхи, алгоритм роботи служби анітрохи не зміниться, але це призведе до неоптимальної витрати дискового простору;

– при копіюванні VM, заснованої на розділах, а не на файлах віртуальних жорстких дисків, необхідно попередньо подбати про те, щоб створити новий розділ для її розміщення, і або використовувати для цих цілей реальний розділ, або скористатися будь-якою SAN-технологією, наприклад, iSCSI, для створення логічного розділу, заснованого на образі.

3.4.4. Міграція існуючих VM

Даний процес вирішує основну задачу даної роботи – централізоване управління міграцією VM. Для детального опису конкретних команд Xen і налаштувань вузлів, які необхідно здійснити для виконання міграції, рекомендується звернутися до [9], а також до відповідних розділів [11] і [12].

Робота даного процесу пов'язана головним чином з керуванням виконанням команд Xen, таких як, наприклад, `xm migrate`, безпосередньо запускає міграцію. Алгоритм виконання процесу наступний:

– адміністратор вибирає VM, яка підлягає міграції, за допомогою веб-інтерфейсу системи - вказуються також цільовий вузол і стратегія міграції (див. п. 2.2);

– система перевіряє, чи не розміщена ця VM на файлової систем, що є загальною для вихідного і цільового вузлів розміщення;

– якщо в результаті перевірки в пункті 2 було отримано значення «істина», то виконується міграція - з використанням штатних засобів Xen і відповідно до обраної в пункті списку 1 стратегією, і алгоритм завершує роботу;

– якщо між вузлами немає загальної файлової системи, то «жива» міграція неможлива, і адміністратор попереджається про це, після чого пропонується здійснити «холодну» міграцію замість «живої». Якщо «холодна» міграція не є прийнятним варіантом, то алгоритм завершує роботу;

– якщо адміністратором був зроблено вибір на користь «холодної» міграції, то здійснюються наступні кроки: збереження стану VM і її припинення; читання

файлу конфігурації VM; якщо VM має віртуальні жорсткі диски, засновані на файлах, то проводиться виділення необхідного дискового простору на цільовому вузлі, після чого ці файли синхронізуються за допомогою утиліти `rsync`, далі виконується перехід через один підпункт; якщо VM має віртуальні жорсткі диски, що ґрунтуються на розділах, то вони синхронізуються один з одним по протоколу SSH; редагування файлу конфігурації VM; передача збереженого стану VM на цільовий вузол; «холодна» міграція VM здійснюється штатними засобами Xen;

- координатор сповіщається про успішну міграцію VM;
- запуск VM.

Як видно з опису алгоритму, найбільш складною є ситуація, коли необхідно провести «холодну» міграцію VM при відсутності загального мережевого сховища. При виникненні будь-яких помилок в ході виконання алгоритму адміністратор повідомляється про них.

На рис. 3.10 представлена блок-схема цього алгоритму.

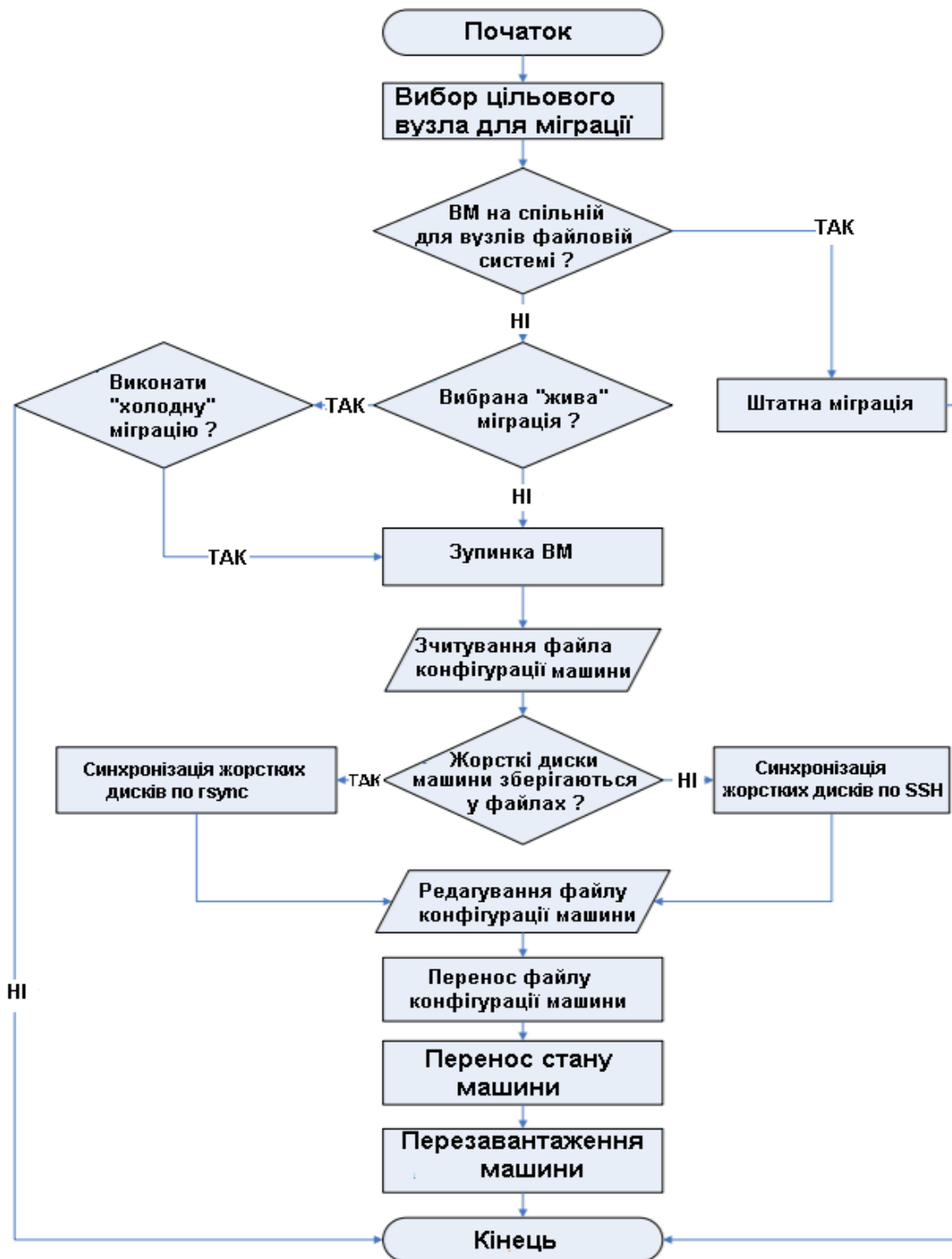


Рис. 3.10. Блок-схема алгоритму міграції VM

Даний процес, як і описаний в п. 3.4.3 процес розгортання нових екземплярів VM на основі шаблону, в основному, базується на низькорівневих операціях, і тому майже не використовує можливості ООП. Однак, для комунікації між вузлами, як і в разі процесу розгортання VM на основі шаблону, застосовується протокол НТТР, а також використовується вбудований НТТР-сервер агента, оскільки даним процесом використовуються ті ж класи для роботи з цим протоколом

Крім згаданих налаштувань HTTP-сервера, даний процес не потребує додаткових налаштувань, оскільки всі необхідні дані для координації його виконання надходять разом з HTTP-запитами.

При пошуку варіантів цільового вузла міграції VM використовується представлення VM, представлення БД `view_migration_options` таблиця `vm_resources_need`, а також вже згадувані раніше таблиці `host`, `stats_average_period` і представлення `view_host_health`.

Таблиця `vm_resources_need` містить записи про потреби VM в обчислювальних ресурсах - ці дані використовуються при пошуку найбільш оптимально придатних для міграції вузлів, та її поля мають таке ж значення, що і відповідні поля таблиці `update`. Якщо в цій таблиці немає запису для будь-якої VM, то вважається, що VM не пред'являє мінімальних вимог до ресурсів вузла. Якщо ж запис існує, але одне або кілька її полів мають значення 0 або NULL, то вважається, що VM не пред'являє мінімальних вимог до відповідного ресурсу вузла.

При формуванні представлення `view_migration_options` для кожної VM проводиться пошук всіх активних вузлів хмари, у яких середня завантаженість ресурсів за останні 5 хвилин, яку видобувають із таблиці `stats_average_period`, задовольняє вимоги, представлені відповідним записом таблиці `vm_resources_need`. Крім цих, кількісних показників проводиться перевірка статусу вузла за допомогою представлення `view_host_health` - в якості можливих варіантів міграції пропонуються вузли, у яких немає перевантажених ресурсів.

Можливість автоматичної міграції на даний момент ще не реалізована в системі, але зрозуміло, що при наявності певної системи прийняття рішень про необхідність міграції цей процес можна було б виконати. Передбачається, що прийняття рішення про необхідність міграції якоїсь VM здійснюється на основі завантаженості вузла, на якому вона знаходиться зараз, і у відповідності з якоюсь стратегією використання ресурсів хмари. Можна виділити дві таких стратегії:

- стратегія максимальною щільності обчислювальних процесів - метою цієї стратегії є максимальне завантаження окремих вузлів, таким чином досягається максимальна густина обчислювальних процесів на вузол;

– стратегія максимального розсіювання обчислювальних процесів – дана стратегія протилежна попередній, і її завдання полягає в тому, щоб мінімізувати завантаження окремих вузлів, розподіляючи ВМ по вузлах таким чином, щоб середня завантаженість вузла хмари зростала мінімально повільно.

Існують обмеження, які необхідно враховувати:

– як і в випадку процесу розгортання ВМ на основі шаблону, даний процес в разі «холодної» міграції без загального файлового сховища може бути вельми розтягнутий у часі - це пов'язано з великим об'ємом віртуальних жорстких дисків;

– при «холодній» міграції ВМ, заснованої на розділах, необхідно заздалегідь потурбуватися про те, щоб відповідний розділ цільового вузла існував і мав в точності той же самий розмір, що і на вихідному вузлі.

3.5. Висновки до розділу

У цьому розділі описано функціональні можливості та архітектура розробленого програмного засобу Xen Migr. Наведена програмна архітектура розробки. Описана організація роботи системних служб. Докладно описано і проілюстровано основні процеси в хмарі, в т.ч. міграція існуючих ВМ.

РОЗДІЛ 4

ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Метою цього розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності від дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі, а також прийняття рішення щодо подальшого розвитку і впровадження або ж недоцільність впровадження відповідної розробки.

Для здійснення оцінки потрібно зробити розрахунки трудомісткості кожної операції, що мала місце при проведенні наукових досліджень.

4.1. Розрахунок норм часу на виконання науково-дослідної роботи

Реалізація проекту дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі складається з низки послідовних та взаємопов'язаних етапів. Кожен із етапів реалізації проекту характеризується метою та змістом, оцінкою часу виконання, кількістю та спеціалізацією виконавців, а також приблизною оцінкою вартості.

Реалізація проекту складається із підготовчого етапу, етапу технічної пропозиції, створення технічного завдання, проектування системи, практичної реалізації, тестування, верифікації та заключного етапу.

Норми часу на виконання науково-дослідницької роботи розраховуватимуться на основі середнього часу виконання стадії в годинах, що наведені в табл. 4.1 разом із інформацією про виконавців і сумарною кількості затраченого часу.

Операції технологічного процесу та їх час виконання

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Підготовча стадія	Проектний менеджер	10
		Інженер-програміст	
2	Технічна пропозиція	Проектний менеджер	10
		Інженер-програміст	
3	Створення технічного завдання	Проектний менеджер	20
		Інженер-програміст	
4	Проектування системи	Інженер-програміст	20
5	Практична реалізація	Інженер-програміст	135
6	Тестування системи	Тестувальник	20
7	Верифікація системи	Тестувальник	20
		Інженер-програміст	
		Проектний менеджер	
8	Створення документації	Інженер-програміст	20
9	Заключна стадія	Проектний менеджер	10
Разом			265

В підсумку на реалізацію проекту дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі необхідно 265 людино-годин, залучення трьох спеціалістів та виконання дев'яти різноманітних стадій реалізації проекту.

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

Визначення витрат на оплату праці та відрахувань на соціальні заходи прямо залежить від кількості витраченого працівниками часу на роботу, ставки в годину чи місяць, кількість відрахувань на соціальні заходи встановлених в законному порядку на час розрахунку.

В результаті розрахунку потрібно визначити основну та додаткову заробітну плату, витрати на соціальні заходи та на основі цих даних визначити сумарні витрати на оплату праці. Основна заробітна плата нараховується за виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами. Додаткова заробітна плата – це складова заробітної плати працівників,

до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час.

При розрахунку заробітної плати кількість робочих днів у місяці слід в середньому приймати – 24,5 дні/міс., або ж 196 год./міс. (тривалість робочого дня – 8 год.).

Наймані працівники для дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі працюють згідно контракту, який в якому вказано їхню погодинну ставку. Тобто розрахунок заробітної плати працівників відбуватиметься на базі тарифної ставки та кількості відпрацьованих годин.

У штаті найманих працівників для дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі залучено проектного менеджера, інженера-програміста і тестувальника.

Тарифні ставки учасників процесу дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі:

- Проектний менеджер – 150 грн./год.
- Інженер-програміст – 130 грн./год.
- Тестувальник – 100 грн./год.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c \cdot K_r, \quad (4.1)$$

де T_c – тарифна ставка, грн.; K_r – кількість відпрацьованих годин.

Оскільки всі види робіт в виконує три спеціаліста, то основна заробітна плата буде розраховуватись за даною формулою (4.1):

$$Z_{\text{осн.}} = 150 \cdot 35 + 130 \cdot 200 + 100 \cdot 30 = 34250 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати й визначається за формулою (4.2).

Коефіцієнт додаткових виплат працівникам становить 0,1.

$$З_{\text{дод.}} = З_{\text{осн.}} \cdot K_{\text{допл.}}, \quad (4.2)$$

де $K_{\text{допл}}$ – коефіцієнт додаткових виплат працівникам

$$З_{\text{дод.}} = 34250 \cdot 0,1 = 3425 \text{ грн.}$$

Звідси загальні витрати на оплату праці (фонд заробітної плати) визначаються за формулою (4.3):

$$В_{\text{о.п.}} = З_{\text{осн.}} + З_{\text{дод.}} \quad (4.3)$$

$$В_{\text{о.п.}} = 34250 + 3425 = 37675 \text{ грн.}$$

З цієї суми утримуються обов'язкові відрахування на соціальні заходи:

- Єдиний соціальний внесок, що становить 22%;
- Військовий збір, що становить 1,5%;
- податок на доходи фізичних осіб: 18%;

Сума відрахувань становить 41,5% від фонду оплати праці та визначається за формулою:

$$В_{\text{с.з.}} = \Phi_{\text{оп}} \cdot 0,415 \quad (4.4)$$

де $\Phi_{\text{оп}}$ – фонд оплати праці, грн.

$$В_{\text{с.з.}} = 37675 \cdot 0,415 = 15635,125$$

Усі витрати обчислюються детально наведені в табл. 4.2 та обчислюються за формулою:

$$В_{\text{зп}} = \Phi_{\text{ЗП}} + \Phi_{\text{ОП}} \quad (4.5)$$

$$В_{\text{зп}} = 34250 + 15635,125 = 49885,125 \text{ грн.}$$

Розрахунки витрат на оплату праці

з/ п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на ФОП, грн.	Всього витрати на плату праці, грн. (6=3+4+5)
		Тарифна ставка, грн.	Кількість відпрацьованих год.	Фактично нарах. з/пл., грн.			
1.	Проектний менеджер	150	35	5250	525	-	-
2.	Інженер-програміст	130	200	26000	2600	-	-
3.	Тестувальник	100	30	3000	300	-	-
Разом		380	265	34250	3425	15635,125	49885,125

Опираючись на розрахунки витрат на оплату та таблицю результатів 4.2 видно, що всього витрати на плату праці становлять 49885,125 грн.

4.3. Розрахунок матеріальних витрат

Матеріальні витрати є невід'ємною частиною розробки та визначаються як добуток кількості витрачених матеріалів та їх ціни за формулою:

$$M_{ei} = q_i \cdot p_i, \quad (4.6)$$

де: q_i – кількість витраченого матеріалу i -го виду; p_i – ціна матеріалу i -го виду.

Звідси, загальні матеріальні витрати можна визначити за формулою:

$$З_{м.в.} = \sum M_{ei}. \quad (4.7)$$

Результати проведених розрахунків наведено у табл. 4.3.

Таблиця 4.3

Результати розрахунків матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Фактично витрачено матеріалів	Ціна одиниці, грн.	Загальна сума витрат, грн.
1	CD диски	шт.	2	7,45	14,90
2	Папір для друку	листів	500	0,15	75,00
3	Чорнила для принтера	шт.	1	80,00	80,00
Всього					169,90

Згідно проведених розрахунків, матеріальні витрати становлять 169,90 грн.

4.4. Розрахунок витрат на електроенергію

Однією із статей витрат є витрати на електроенергію під час проходження усіх етапів реалізації кінцевого продукту.

Затрати на електроенергію одиниці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S, \quad (4.8)$$

де W – необхідна потужність, кВт; T – кількість годин на реалізацію розробки; S – вартість кіловат-години електроенергії.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів. Отже, 1 кВт з ПДВ коштує 2,42 грн.

Потужність комп'ютерів для реалізації кінцевого продукту – 400 Вт, кількість годин роботи обладнання згідно таблиці 4.1 – 265 годин.

Визначимо витрати на електроенергію згідно формули :

$$Z_e = 0,4 \cdot 265 \cdot 2,42 = 256,52 \text{ грн.}$$

Згідно формули затрати на електроенергію становлять 256,52 грн.

4.5. Розрахунок суми амортизаційних відрахувань

Для будь якої діяльності характерною є властивість зношування на зниження якості властивостей інструментарію та фондів за допомогою яких ведеться діяльність. Для вирішення проблеми із відновленням даних фондів використовується амортизація, що являє собою процес трансформації вартості основних фондів на вартість продукції, яка щойно була створена, задля повного відновлення основних фондів.

Для визначення амортизаційних відрахувань використовується формула:

$$A = \frac{B_B \cdot H_A}{100\%} \quad (4.9)$$

де A – амортизаційні відрахування за звітний період, грн.; B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.; H_A – норма амортизації.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Річний робочий фонд становитиме 2352 годин, так як робочий день становить 8 годин, а кількість робочих днів в місяці становить 24,5 годин.

Для даної розробки засобом розробки є комп'ютер. Його сума становить 18500 грн. Отже, амортизаційні відрахування будуть рівні:

$$A = 18500 \cdot 5\% / 100\% = 925 \text{ грн.}$$

Згідно проведених обчислень амортизаційні відрахування становлять 925 грн.

4.6. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління спілкою та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_e = B_{o.n.} \cdot 0,2 \dots 0,6, \quad (4.10)$$

де H_e – накладні витрати.

Отже, накладні витрати становлять згідно формули (4.10):

$$H_e = 37675 \cdot 0,2 = 7535 \text{ грн.}$$

Накладні витрати згідно розрахунку формули, становить 7535 грн.

4.7. Складання кошторису витрат та визначення собівартості науково-дослідницької роботи

Результати проведених вище розрахунків наведено у табл. 4.4.

Таблиця 4.4

Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці	43103,6	63,74
Відрахування на соціальні заходи	15635,125	23,12
Матеріальні витрати	169,9	0,25
Витрати на електроенергію	256,52	0,38
Амортизаційні відрахування	925	1,37
Накладні витрати	7535	11,14
Собівартість	67625,145	100,00

Собівартість (C_e) програмного продукту розраховуємо за формулою:

$$C_{\epsilon} = B_{з.п.} + B_{с.з.} + З_{м.в.} + З_{\epsilon} + A + H_{\epsilon} . \quad (4.11)$$

Отже, собівартість програмного продукту дорівнює:

$$C_B = 49885,125 + 15635,125 + 169,90 + 256,52 + 925 + 7535 = 74406,67 \text{ грн.}$$

Загальний кошторис витрат та визначення собівартості науково-дослідницької роботи становить 74406,67 грн.

4.8. Розрахунок ціни програмного продукту

Ціну науково-дослідної роботи можна визначити за формулою:

$$Ц = \frac{C_B \cdot (1 + P_{рен.}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \quad (4.12)$$

де $P_{рен.}$ – рівень рентабельності (30 %); K – кількість замовлень, од. (встановлюється лише при розробці програмного продукту та мікропроцесорних систем); $B_{н.і.}$ – вартість носія інформації, грн. (встановлюється лише при розробці програмного продукту); $ПДВ$ – ставка податку на додану вартість (20%).

Оскільки розробка є прикладною, і використовуватиметься тільки для одного підприємства, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{н.і.}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$Ц = C_B \cdot (1 + P_{рен.}) \cdot (1 + ПДВ) \quad (4.13)$$

Звідси ціна на роботу складе:

$$Ц = 74406,67 \cdot (1 + 0,3) \cdot (1 + 0,2) = 116074,4 \text{ грн.}$$

Загальний розрахунок ціни програмного продукту становить 116074,4 грн.

4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \frac{П}{C_B}, \quad (4.14)$$

де $П$ – прибуток; C_B – собівартість.

Плановий прибуток ($П_{пл}$) знаходимо за формулою:

$$П_{пл} = Ц - C_v. \quad (4.15)$$

Розраховуємо плановий прибуток:

$$П_{пл} = 116074,4 - 74406,67 = 41667,73 \text{ грн.}$$

Отже, формула для визначення економічної ефективності набуде вигляду:

$$E_p = \frac{П_{пл}}{C_B}. \quad (4.16)$$

Тоді,

$$E_p = 41667,73 / 74406,67 = 0,56.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p}, \quad (4.17)$$

Термін окупності дорівнює:

$$T_p = 1 / 0,5 = 1,78 \text{ р.}$$

Згідно формул плановий прибуток від розробки становить 41667,73 грн., економічна ефективність дорівнює 0,56, а термін окупності становить 1,78 роки що вважається доцільним та економічно вигідним.

4.10. Висновки до розділу

В організаційно-економічній частині дипломної роботи освітнього рівня «магістр» було розраховано основні техніко-економічні показники дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі (див. таблиця 4.5).

Таблиця 4.5

Техніко-економічні показники науково-дослідної роботи

№ п/п	Показник	Значення
1	Собівартість, грн.	74406,67
2	Плановий прибуток, грн.	41667,73
3	Ціна, грн.	116074,4
4	Економічна ефективність	0,56
5	Термін окупності, рік	1,78

Орієнтоване значення економічної ефективності становить 0,56, що є достатньо високим значенням.

Період окупності повинен варіюватися від 1 до 3 років, тоді розвиток вважається доцільним та економічно вигідним. Термін окупності даної роботи становить 1,78 років.

На основі проведених обрахунків можна зробити висновок, що дослідження і використання методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі є доцільним у зв'язку з невеликим терміном окупності та великим обсягом планового прибутку.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

Метою дипломної роботи магістра є дослідження методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі. Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників з розробки ПЗ комп'ютерних систем, в тому числі і фахівців з дослідження методів та інструментальних засобів, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [18].

Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Згідно Вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до цих вимог;

– переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 22.08.2005 N 161, зареєстрованого в Міністерстві юстиції України 05.09.2005 за N 990/11270 (НАПБ Б.06.004-2005);

– Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду

України від 28.10.98 N 247 (далі - ДБН В.2.5-56:2014, з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014.

Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, ДБН В.2.5-56:2010, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним.

Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі.

Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При експлуатації інструментального засобу управління процесом міграції віртуальних машин в обчислювальній хмарі, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмар повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги».

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп'ютера,

зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців з дослідження методів та інструментальних засобів управління процесом міграції віртуальних машин в обчислювальній хмарі.

5.2. Оцінка дії електромагнітного імпульсу (ЕМІ) на елементи виробництва та методи захисту

У воєнний час при застосуванні ядерної зброї проти України на електронно-обчислювальне обладнання в першу чергу буде впливати ЕМІ ядерного вибуху у вигляді короткого імпульсу, який вражає головним чином електричну та електронну апаратуру.

ЕМІ виникають в основному в результаті взаємодії гамма-випромінювання з атомами навколишнього середовища. На утворення ЕМІ йде невелика кількість ядерної енергії, але він здатен викликати високі імпульси струмів та напруг в кабелях повітряних і підземних ліній зв'язку, сигналізації, управління, електропередачі, в антенах радіостанцій. Вплив ЕМІ може привести до згорання чутливих електронних та електричних елементів, зв'язаних з великими антенами чи відкритими дротами, а також до порушень в обчислювальних пристроях. Вплив ЕМІ необхідно враховувати для всіх електричних та електронних систем. Для найбільш важливих приладів треба використовувати засоби захисту і підвищувати їх стійкість до ЕМІ.

Особливістю ЕМІ, як вражаючого фактору є його здатність розповсюджуватись на десятки і сотні кілометрів в оточуючому середовищі. Тому ЕМІ може вплинути своєю дією на об'єкти, там де вибухова хвиля, світлове випромінювання, проникаюча радіація втрачають своє значення, як вражаючі фактори. При наземних та низьких повітряних вибухах в лініях зв'язку та електрозабезпечення виникають напруги, які можуть викликати пробій ізоляції провідників та кабелів відносно землі, пробій ізоляції елементів приладів

підключених до повітряних і підземних ліній. Степінь враження залежить від наведеного імпульсу напруги чи струму і також електричної міцності обладнання.

Найбільш піддані впливу ЕМІ системи зв'язку, сигналізації, управління. Використані в цих системах кабелі та апаратура мають обмежену електричну міцність не більше 10кВ імпульсної напруги, тоді як наведені імпульси напруги від ЕМІ можуть перевищувати ці значення. Найбільш піддана впливу ЕМІ апаратура виконана на напівпровідниках та інтегральних схемах, працюючих на малих струмах і напругах, і значить відчутних до впливу зовнішніх електричних і магнітних кіл, в тому числі і елементи програмного засобу для управління процесом міграції віртуальних машин в обчислювальній хмарі. ЕМІ пробиває ізоляцію, спалює елементи електричних схем радіоапаратури, викликає коротке замикання в радіопристроях, іонізацію діелектриків, змінює або повністю стирає магнітний запис. Встановлено, що при дії ЕМІ на апаратуру найбільша напруга наводиться на вході. В транзисторах відбувається така залежність: чим більший коефіцієнт підсилення транзистора, тим менша його електрична міцність.

ЕМІ пошкоджує також резистори, викликає іскріння в їх міжконтактних з'єднаннях і деяких областях провідної поверхні. Найбільшу небезпеку ЕМІ представляє для апаратури, яка встановлена в особливо міцних спорудах, які витримують великі тиски ударної хвилі. В цих спорудах апаратура не виходить з ладу від механічних пошкоджень, але ЕМІ може вивести з ладу всю незахищену апаратуру системи зв'язку, сигналізації і керування. Найбільших значень досягають напруги, які наводяться між кабелем і землею. Напруженість електромагнітного поля всередині споруди в деяких випадках недостатня для того, щоб вивести з ладу апаратуру, але такі поля в змозі викликати короточасний збій роботи радіотехнічних пристроїв.

Розглянемо можливі шляхи рішення задачі захисту від ЕМІ програмного засобу для управління процесом міграції віртуальних машин в обчислювальній хмарі. Ідеальним захистом від ЕМІ виявилось б повне укриття приміщення, в якому розміщена радіоелектронна апаратура, металевим екраном.

Водночас зрозуміло, що практично забезпечити такий захист у ряді випадків неможливо, тому що для роботи апаратури часто потрібно забезпечити її електричний зв'язок із зовнішніми пристроями. Тому використовуються менш

надійні засоби захисту, такі, як струмопровідні сітки, або плівкові покриття для вікон, щільникові металеві конструкції для повітрезабірників і вентиляційних отворів і контактні пружинні прокладки, розміщені по периметру дверей і люків.

Більш складною технічною проблемою рахується захист від проникнення ЕМІ в апаратуру через різноманітні кабельні входи. Радикальним рішенням даної проблеми міг би стати перехід від електричних мереж зв'язку до практично не схильних до впливу ЕМІ волоконно-оптичних. Проте заміна напівпровідникових приладів у всьому спектрі виконуваних ними функцій електронно-оптичними пристроями можлива тільки у віддаленому майбутньому. Тому в даний час в якості засобів захисту кабельних входів найбільш широко використовуються фільтри, у тому числі волоконні, а також іскрові розрядники, металлоокисні варистори і високошвидкісні зенеровські діоди.

Всі ці засоби мають як переваги, так і недоліки. Так, ємнісно-індуктивні фільтри достатньо ефективні для захисту від ЕМІ малої інтенсивності, волоконні фільтри захищають у відносно вузькому діапазоні надвисоких частот. Іскрові розрядники мають значну інерційність й в основному придатні для захисту від перевантажень, що виникають під впливом напруг і струмів, що наводяться в обшивці літака, кожусі апаратури й оплетенні кабеля.

Металлоокисні варистори є напівпровідниковими приладами, що різко підвищують свою провідність при високій напрузі. Проте, при застосуванні цих приладів у якості засобів захисту від ЕМІ варто враховувати їх недостатньо високу швидкодію і погіршення характеристик при кількаразовому впливі навантажень. Ці недоліки відсутні у високошвидкісних зенеровських діодах, дія яких заснована на різкій лавиноподібній зміні опору від високого значення практично до нуля, при перевищенні прикладеної до них напруги граничного розміру. Крім того на відміну від варисторів характеристики зенеровських діодів після багатократних впливів високих напруг і переключень режимів не погіршуються.

Найбільш раціональним підходом до проектування засобів захисту від ЕМІ кабельних входів є створення таких роз'ємів у конструкції яких передбачені спеціальні заходи, що забезпечують формування елементів фільтрів і установку вмонтованих зенеровських діодів. Подібне рішення сприяє одержанню дуже малих значень ємності й індуктивності, що необхідно для забезпечення захисту від

імпульсів, що мають незначну тривалість і, отже, потужну високочастотну складову. Використання роз'ємів подібної конструкції дозволить вирішити проблему обмеження малогабаритних характеристик пристрою захисту.

Складність рішення задачі захисту від ЕМП і висока вартість розроблених для цих цілей засобів і методів змушують піти по шляху їхнього вибіркового застосування в особливо важливих системах зброї і військової техніки. Такий же шлях обраний і для захисту систем, що мають велику протяжність, керування і зв'язку. Проте основним методом рішення даної проблеми спеціалісти вважають створення так званих розподілених мереж зв'язку.

5.3. Врахування шкідливих і небезпечних умов праці персоналу в ході провадження виробничої діяльності суб'єктами господарювання

Умови праці на виробництві диференціюються залежно від фактично визначених рівнів та факторів виробничого середовища порівняно із санітарними нормами, правилами, гігієнічними нормативами, а також з урахуванням їх можливого шкідливого впливу на стан здоров'я працюючих. Шкідливі умови — характеризуються такими рівнями шкідливих виробничих факторів, які перевищують гігієнічні нормативи і здатні чинити несприятливий вплив на організм працюючого та (або) його майбутніх нащадків. Шкідливі умови за показниками перевищення гігієнічних нормативів та вираженості можливих змін в організмі працюючих поділяються на чотири ступеня, де четвертий (найтяжчий) – це небезпечні (екстремальні) умови.

В Україні створена належна база нормативно-правових актів щодо організації і проведення атестації робочих місць за умовами праці, а також з надання працівникам підприємств пільг і компенсацій за особливо важкі та особливо шкідливі також за важкі і шкідливі умови праці. Зокрема Постанова Кабінету Міністрів України від 12.07.2005 р. № 576 «Про затвердження переліку робіт із важкими, шкідливими та особливо шкідливими умовами праці у будівництві, на яких встановлюється підвищена оплата праці», Державні санітарні норми та правила «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та

напруженості трудового процесу» (затверджені наказом Міністерства охорони здоров'я України від 08.04.2014 № 248) та інші документи.

Для врахування шкідливих і небезпечних умов праці персоналу в ході провадження виробничої діяльності суб'єктами господарювання (підприємствами, установами і організаціями незалежно від форм власності і господарювання, де технологічний процес, використовуване обладнання, сировина та матеріали є потенційними джерелами шкідливих і небезпечних виробничих факторів), проводиться атестація робочих місць.

Відповідно до ст. 153 Кодексу законів «Про працю в Україні», на всіх підприємствах, в установах, організаціях створюються безпечні і нешкідливі умови праці. Забезпечення безпечних і нешкідливих умов праці покладається на власника або уповноважений ним орган. Власник або уповноважений ним орган зобов'язаний провести (забезпечити) лабораторні дослідження робочих місць, на яких існують шкідливі і важкі умови праці, встановити пільги і гарантії для працівників, що працюють у цих умовах, та розробити заходи, що забезпечують усунення причин виникнення нещасних випадків і професійних захворювань. Для цього проводиться атестація робочих місць за умовами праці. Керівник підприємства затверджує перелік робочих місць, виробництв, робіт, професій і посад працівників, яким підтверджено право на пільги і компенсації, а також на пільгове пенсійне забезпечення. Шкідливі умови праці суттєво впливають не тільки на виникнення професійних захворювань, а й на виникнення і тривалість загальних захворювань. Відповідно, особи, які будуть працювати на робочих місцях із шкідливими та небезпечними умовами праці, підвищеними фізичними та емоційними навантаженнями насамперед підлягають професійному відбору, попередньому та періодичному медичних оглядах.

Підприємство повинно враховувати витрати на компенсацію за роботу в несприятливих умовах, що не відповідають санітарним нормам (надавати пільги за важкі і шкідливі умови праці), зокрема додаткові відпустки; скорочений робочий день; лікувально-профілактичне харчування; безкоштовна видача молока чи інших рівноцінних продуктів; підвищені тарифні ставки; доплати за умови та інтенсивність праці; пільгові пенсії. Також працівники зі шкідливими умовами праці повинні забезпечуватися спеціальними захисними засоби, такими як:

спецодяг та спецвзуття; засоби захисту рук (рукавиці), захисно-профілактичні засоби (пасти, мазі) та очисники шкіри (мило, синтетичні мийні засоби); засоби індивідуального захисту органів дихання; засоби захисту голови (каска, шоломи); засоби захисту очей і обличчя (захисні окуляри та маски); засоби захисту органу слуху (шоломи, антифони).

5.4. Висновки до розділу

В цьому розділі проаналізовано важливі питання охорони праці та безпеки в надзвичайних ситуаціях, оцінено дію ЕМІ на елементи програмного засобу і методи захисту, розглянуто проблему шкідливих і небезпечних умов праці персоналу в ході роботи з програмних засобом і запропоновано варіанти компенсації за роботу у таких умовах та спеціальні засоби захисту.

РОЗДІЛ 6

ЕКОЛОГІЯ

6.1. Аналіз сучасних програмних продуктів для обробки великих масивів екологічної інформації

Оперативна, якісна і точна обробка великих масивів статистичної інформації може бути виконана лише з використанням сучасних засобів обчислювальної техніки. Наявність потужних, надійних і разом з тим простих в експлуатації програмних продуктів статистичного аналізу звільняє дослідника від рутинних операцій, розширює сферу застосування статистичних методів в різних галузях людської діяльності, сприяє появі якісно нових можливостей статистичного аналізу і моделювання даних. Використання пакетів прикладних програм - це єдиний реальний практичний інструмент розв'язування задач багатофакторного кореляційно-регресійного та аналізу в багатовимірному просторі [19].

Програмне забезпечення статистичних досліджень досить розвинуте. Сучасний ринок програмних продуктів пропонує різноманітні пакети програм для статистичної обробки даних. Всесвітньо відомі статистичні пакети для комплексної обробки даних: BMDP, SPSS, SAS, Systat, Minitab, S-Plus, Statgraphics, Statistica та інші [19].

Використання згаданих пакетів програм дає змогу автоматизувати процес статистичного дослідження в таких напрямках: створення файлів даних і таблиць; групування даних; графічний аналіз даних; розрахунок варіаційних характеристик вибірових сукупностей; побудова рядів розподілу; аналіз рядів динаміки і прогнозування їх майбутніх рівнів; кореляційно-регресійний аналіз; багатомірний аналіз.

З 1995 р. світовим лідером на ринку статистичного програмного забезпечення визнається інтегрована система Statistica для Windows, розроблена фірмою Stat Soft. Перша версія програми з'явилася у 1991р. для операційної системи MS-DOS і була новим напрямом розвитку статистичного програмного забезпечення. В ній реалізовано графічно-орієнтований підхід до статистичного

аналізу даних, суть якого полягає в отриманні всебічного візуального представлення інформації на всіх етапах статистичної обробки даних.

Багатофункціональна, графічно орієнтована на обробку масових даних система Statistica відповідає основним стандартам Windows (динамічний обмін даними з іншими додатками, підтримка основних операцій з буфером обміну, робота в мережевому середовищі та інші).

Передусім це стандарти користувацького інтерфейсу — MDI, використання буфера-обміну, механізму динамічного зв'язку (DDE) з іншими додатками; система підтримує всі операції, реалізовані за допомогою методу Drag-and-Drop — «Перетягти та опустити», включаючи автозаповнення, інші.

Система Stratgraphics працює з чотирма типами документів. Це: електронна таблиця Spreadsheet, призначена для введення і перетворення первинних даних; електронна таблиця Scrollsheet — для виведення результатів аналізу; графік — для візуалізації результатів обробки та аналізу даних; звіт — файл у формі RTF (розширений текстовий формат), в якому зберігається текстова, числова і графічна інформація.

Усі статистичні процедури системи розбито на окремі модулі, кожен з яких об'єднує групу логічно зв'язаних між собою статистичних методів і в рамках конкретної моделі забезпечує повний і всебічний аналіз закономірностей.

Використання сучасних комп'ютерних технологій обробки даних, інтерактивний спосіб взаємодії з системою перетворюють статистичний аналіз, моделювання та прогнозування в захоплююче дослідження закономірностей навколишнього світу. Завдяки різноманітним формам організації діалогу, максимально простій із звичними для статистики термінами мові спілкування, наявності контекстно-залежної довідкової системи, мові програмування STATISTICA BASIC пакет є ефективним інструментом проведення статистичного дослідження як для користувача-початківця, так і для професіонала.

Statistica - це відносно дорогі, а отже, і не завжди доступні програмні продукти. Переважну більшість задач статистичного аналізу можна досить ефективно розв'язувати з використанням програми обробки електронних таблиць Microsoft Excel. Спектр доступних стандартних статистичних функцій останніх версій Microsoft Excel майже не поступається спеціальним програмам обробки

статистичних даних (понад 70 функцій).

6.2. Статистична оцінка техногенних впливів

Така оцінка містить наступні складові: комплексна оцінка техногенного впливу на біосферу; комплексна оцінка території; оцінка впливу на людину. Розглянемо ці складові частини окремо [19].

Комплексна оцінка техногенного впливу на біосферу. Процес незворотного перетворення людиною частин біосфери на техногенні об'єкти і території дістав назву техногенезу, а частина біосфери, штучно перетворена в результаті життєдіяльності людини і заповнена її продуктами, називається технічною оболонкою біосфери (техносферою). Основні завдання цієї оцінки: вивчення техногенних чинників забруднення довкілля; класифікація джерел забруднень; визначення джерел походження забруднень; всі проблемами людства, які породжують забруднення біосфери. Техногенні чинники забруднення довкілля об'єднують у такі групи: атмосферні - хімічне, фізичне, механічне і теплове забруднення; водні - океани і моря, забруднення поверхневих і підземних вод; ґрунтові - хімічне, ерозійне забруднення, ущільнення, засолення, заболочення тощо; геологічні - негативні екзогенні процеси - зсуви, підтоплення, обвали, абразії берегів тощо; біотичні - деградації екосистем, збіднення біорізноманіття, мутації, зникнення лісів і пасовищ, біогенна акумуляція шкідливих речовин тощо; комплексні - порушення природної структури ландшафтів, поява пустель, деградація земель.

Забруднення класифікують за галузевим принципом: промислові - хімічна промисловість, металургійна, видобувна тощо; транспортні - автотранспорт, авіаційний, морський тощо; енергетичні - теплові і атомні електростанції; сільськогосподарські - засоби захисту рослин, мінеральні та органічні добрива тощо; пов'язані з військовою діяльністю. Вплив техносфери на стан атмосфери: найбільший вплив на стан атмосфери чинять теплоенергетика, металургійна промисловість, підприємства хімічної та будівельної індустрії, автотранспорт, що викидають у повітря пил, важкі метали, вуглеводні, оксиди карбону, бенз(а)пірен та інші речовини; найбільший вплив на хімічний склад атмосферного повітря

чинить спалювання кам'яного вугілля; найпотужнішим негативним техногенним чинником є енергетика - підприємства чорної металургії утворюються пил та оксид сірки, хімічна і нафтохімічна промисловість продукують майже у два рази менше викидів при значно більшій різноманітності забруднюючих речовин; крім газоподібних речовин у повітря потрапляють рідкі і тверді частинки у вигляді аерозолів; серед усіх видів транспорту автомобільний посідає перше місце за кількістю і різноманітністю забруднюючих речовин, а також за кількістю незворотних змін ландшафтів та інших негативних впливів на довкілля. У містах з розвинутою промисловістю внесок автотранспорту в забруднення довкілля досягає 80% усіх забруднень. Проблеми, пов'язані з гідросферою, зумовлені нестачею прісної води для потреб людства, її забрудненням, порушенням природних кругообігів та зменшенням продуктивності водних екосистем. Найбільшими забрудниками водних ресурсів є промисловість, комунальне і сільське господарства країни, які в структурі забруднення водних ресурсів України складають стосовно 60, 20 і 17%. Важливою проблемою України є також забруднення підземних вод.

Комплексна оцінка території. Суть такої оцінки полягає в дослідженні просторової структури історично складених природно-територіальних комплексів та проведенні на цій основі розділу території країни (районуванню) на природні зони (області), округи та райони. Основне завдання комплексної оцінки території в конкретних регіонах: виявлення комплексу несприятливих факторів, що складають необхідний вихідний матеріал для прогнозування можливих негативних наслідків господарської діяльності; визначення характеру і масштабів наслідків; виявлення причини на основі встановлення причинно-наслідкових зв'язків; розробці заходів, спрямованих на ліквідацію, попередженні і компенсації цих наслідків. Основною метою комплексної оцінки території є встановлення суспільної значимості наслідків за існуючих масштабів господарського впливу на рівновагу екосистем.

Процедура комплексної оцінки території ґрунтується на вивченні механізму взаємодії в системі «населення - господарство - природні системи». Ця оцінка спрямована на вивчення: спричиненого діяльністю людини впливу на природні екосистеми регіону; змін у природних екосистемах під впливом цієї діяльності;

наслідків впливу змінених природних систем на суспільство і економіку в цілому.

Існують різні підходи до екологічної оцінки територій: економічна оцінка; соціальна оцінка; природно-ресурсна оцінка. Для отримання екологічних оцінок застосовуються різні моделі: блокові; матричні; картографічні; статистичні. Для деяких видів залежностей відомі моделі, що адекватно їх описують: закон логарифмічно нормального розподілу, закон дифузії та ін. Більшість зв'язків потребує виявлення виду залежностей і значень коефіцієнтів, що часто мають регіональний характер.

Джерелами впливу на природні екосистеми є господарство та населення, які можуть аналізуватися на трьох територіальних рівнях: мікрорівень - окремі виробничі та сільськогосподарські підприємства або їх підрозділи, окремі об'єкти міського господарства або функціональні зони населених пунктів; мезорівень - промислові пункти, центри або вузли, великі сільськогосподарські підприємства, міські та сільські поселення; макрорівень - промислові та сільськогосподарські райони, агломерації, територіально-виробничі комплекси. Головною метою моделювання для отримання екологічних оцінок є послідовне взаємопов'язане дослідження процесу «взаємодія - зміна – наслідок».

Залежно від показника виміру для екологічної оцінки території застосовується ряд кількісних і якісних методів оцінки: якісна бальна оцінка; кількісна оцінка на основі натуральних (абсолютних і відносних) показників; кількісна оцінка на основі вартісних показників.

Оцінка впливу на людину. Види негативного впливу на організм людини умовно можна об'єднати у дві групи: процеси прямого впливу і процеси непрямого впливу. Процеси прямого впливу обумовлені безпосереднім контактом людини з техногенними об'єктами (механізмами, машинами) або робочими агентами цих об'єктів (високою температурою, токсичними речовинами, електричним струмом, електромагнітними полями чи іншими формами енергетичного впливу, активними біологічними організмами, ін.), що можуть завдавати шкоди здоров'ю людини або навіть призводити до її загибелі. Процеси непрямого впливу на організм людини пов'язані з погіршенням умов життя і діяльності людини (склад повітря, температура, вологість, ін.), які зумовлюють процеси метаболізму в організмі людини. Погіршення якості їжі і питної води є

однією з найбільш небезпечних форм непрямого впливу. Це пояснюється чутливістю організму до процесів інтоксикації продуктів, у першу чергу тих, що відповідають за стан метаболізму в організмі людини. Зниження інформаційної цінності природних систем, на відміну від попереднього виду впливу, діє не на організм людини, а на її особистісні характеристики. Повноцінне формування особистості людини може відбуватися тільки на тлі інформаційного контакту з природними системами. Інформаційне руйнування природних систем також негативно впливає на психологічний стан людини

6.3. Висновки до розділу

В цьому розділі проведено аналіз сучасних програмних продуктів для обробки великих масивів екологічної інформації та розглянуто статистичну оцінку техногенних впливів.

ВИСНОВКИ

У процесі виконання дипломної роботи розглянуто поняття віртуалізації, історія розвитку, проведено огляд технологій віртуалізації. Основну увагу приділено аналізу процесу віртуалізації на основі гіпервізора (паравіртуалізації). Наведено основні переваги та недоліки віртуалізації. Досліджені основні типи гіпервізорів. Проаналізовано основні функції та можливості найбільш вживаних гіпервізорів (Xen, VMware ESX Server, Hyper-V, KVM). Досліджено особливості міграції VM в обчислювальній хмарі.

На основі проведеного аналізу спроектовано архітектуру, сформульовано функціональні можливості та програмно реалізовано програмний засіб Xen Migr. Обґрунтована програмна архітектура розробки. Описана організація роботи системних служб. Докладно описано і проілюстровано основні процеси в хмарі, в т.ч. міграція існуючих VM.

Створений в результаті виконання цієї роботи програмний засіб має такі особливості:

- базується на поширених відкритих протоколах, стандартах та технологіях;
- дозволяє без наявності загального мережевого сховища здійснювати міграцію VM;
- можливість задання порогів використання ресурсів вузлами мережі для спостереження за їх станом;
- передбачено автоматичний пошук варіантів цільового вузла при необхідності міграції VM відповідно до потреб останньої.

Подальшим розвитком проекту може бути:

- автоматизація міграції віртуальних машин в хмарі;
- розробка окремого протоколу для комунікації між вузлами засобу;
- дослідження можливостей міграції віртуальних машин між IP-підмережами і / або за відсутності загальних файлових сховищ;
- розробка додаткових функцій для віддаленого управління ресурсами вузлів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Савельев А.О. Решения Microsoft для виртуализации ИТ-инфраструктуры предприятий. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. 285 с.
2. History of virtualization, virtual machines, server consolidation. VMware, Inc., Palo Alto, California, USA. 2010. URL: <http://www.vmware.com> (дата звернення: 05.11.2019).
3. Технологии аппаратной виртуализации. URL: <http://www.ixbt.com/cm/virtualization-h.shtml> (дата звернення: 12.11.2019).
4. Рівні привілеїв гіпервізора. URL: http://upload.wikimedia.org/wikipedia/en/thumb/2/2f/Priv_rings.svg/633pxPriv_rings.svg.png (дата звернення: 10.11.2019).
5. Введение в виртуализацию. Обзорный курс. URL: http://www.pcmag.ru/elearning/course/index.php?COURSE_ID=14 (дата звернення: 12.11.2019).
6. Christopher Clark. Live Migration of Virtual Machines. Cambridge, England, UK : University of Cambridge Computer Laboratory, 2005. 14 p
7. Takemura C., Crawford Luke S. The Book of Xen – A Practical Guide for the System Administrator. San Francisco, California, USA : No Starch Press, 2010. 281 p.
8. О гипервизорах, виртуализации систем и о том, как это работает в облачной среде. URL: <https://www.ibm.com/developerworks/ru/library/clhypervisorcompare/index.html> (дата звернення: 12.11.2019).
9. Вся правда о гипервизорах. URL: <https://www.sim-networks.com/blog/hypervisors-vmware-kvm-xen-openssl> (дата звернення: 12.11.2019).
10. Погорілий С.Д., Білоконь І.В., Бойко Ю.В. Технологія віртуалізації. Динамічна реконфігурація ресурсів обчислювального кластера. *Математичні машини і системи*. 2012. № 3. С. 3-18

11. Von Hagen, W. Professional Xen Virtualization. Indianapolis, Indiana, USA : Wiley Publishing, 2008. 405 p.
12. Paul Barham. Xen and the Art of Virtualization. Cambridge, England, UK : University of Cambridge Computer Laboratory, 2003. 14 p.
13. Hyper-V technology. URL: <https://hyperv.veeam.com/blog/what-is-hyper-v-technology> (дата звернення: 19.11.2019).
14. VMware vCenter Server Capabilities. URL: <https://blogs.vmware.com/vsphere/2013/04/did-you-know-vcenter-servercan-manage-multiple-hypervisors.html> (дата звернення: 19.11.2019).
15. ConVirt centralized view. URL: http://www.convirture.com/media_gallery.php?gallery=6&screen=3 (дата звернення: 19.11.2019).
16. KVM guest OS support. URL: https://www.linux-kvm.org/page/Guest_Support_Status (дата звернення: 19.11.2019).
17. Северин Д.А. Програмний засіб для управління процесом міграції віртуальних машин в обчислювальній хмарі. *Інформаційні моделі, системи та технології*: Праці VII наук.-техн. конф. (Тернопіль, 11-12 грудня 2019 р.) Тернопіль, 2019. С. 96.
18. Зеркалов Д.В. Охорона праці в галузі: Загальні вимоги. Навчальний посібник. К.: «Основа». 2011. 551 с.
19. Тарасова В.В. Екологічна статистика. Підручник. Київ: Центр учбової літератури, 2008. 392 с.

Додаток А
Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



11–12 грудня 2019 року

ТЕРНОПІЛЬ
2019

ПРОГРАМНИЙ ЗАСІБ ДЛЯ УПРАВЛІННЯ ПРОЦЕСОМ МІГРАЦІЇ ВІРТУАЛЬНИХ МАШИН В ОБЧИСЛЮВАЛЬНІЙ ХМАРІ

Одним з сучасних підходів до організації корпоративної інформаційної інфраструктури є віртуалізація обчислювальних систем. Технологія віртуалізації дозволяє значно покращити процес побудови хмарних сервісів, оскільки зменшує кількість фізичного обладнання, вартість на купівлю і обслуговування якого може бути досить високою. Варто зауважити, що існують проблеми з переходом на хмарні технології, наприклад робота сервісів та програмного забезпечення від різних виробників для різних платформ. Для вирішення цих проблем застосовуються технологія міграції. Необхідно забезпечити грамотне керування процесом міграції, завдяки чому можна буде переносити практично будь-яке навантаження x86 у хмару.

На початковому етапі автором розглянуто поняття віртуалізації, історія розвитку, проведено огляд технологій віртуалізації. Основну увагу приділено аналізу процесу віртуалізації на основі гіпервізора (паравіртуалізації). Досліджені основні типи гіпервізорів. Проаналізовано основні функції та можливості найбільш вживаних гіпервізорів (Xen, VMware ESX Server, Hyper-V, KVM). Досліджено особливості міграції віртуальних машин (VM) в обчислювальній хмарі.

На основі проведеного аналізу запропоновано архітектуру (з двома видами серверів - координатор, його роль виконує рівно один вузол, і довільна кількість серверів-агентів), сформульовано функціональні можливості (окремо для координатора та агентів) програмного засобу, який призначений для використання в обчислювальній хмарі. Його вузли функціонують під керуванням гіпервізора Xen [1]. Основним завдання є надання адміністратору можливостей централізованого управління міграцією VM між вузлами хмари. Крім цього, інструментарій розробки включає в себе: засоби збору і відображення статистики

використання обчислювальних ресурсів вузлами мережі і VM; складову управління файлами конфігурацій VM, яка працює разом із системою контролю версій; службу спостереження за станом вузлів хмари, що попереджає адміністратора мережі при падінні кількості вільних ресурсів на вузлах, нижче заданого порогу, або при відмові вузлів; засоби розгортання на вузлах хмари нових екземплярів VM, заснованих на шаблоні.

Визначено процеси, що відбуваються в хмарі, ініційовані і керовані програмним засобом: збір і обробка статистики використання ресурсів обчислювальних вузлів і VM; спостереження за станом вузлів і при необхідності видача попереджень; розгортання нових екземплярів VM на основі шаблону; міграція VM. Обґрунтована програмна архітектура розробки. Докладно описано організацію роботи системних служб та основні процеси в хмарі, в т.ч. міграція існуючих VM. Складові розробки можна розділити на дві групи - системні служби координатора і агентів (розроблені з використанням мови програмування Java і СКБД MySQL) та компоненти для користувача інтерфейсу (створені засобами мови програмування PHP і її бібліотек).

Створений програмний засіб базується на поширених відкритих протоколах, стандартах та технологіях та дозволяє здійснювати міграцію VM без наявності загального мережевого сховища даних.

Література

1. Von Hagen, W. Professional Xen Virtualization / William von Hagen. – Indianapolis, Indiana, USA : Wiley Publishing, 2008. – 405 p.

Додаток Б. Посібник користувача

Даний посібник містить рекомендації по встановленню та налагодженню служб системи на вузли хмари.

Для роботи його служб координатора потрібна СУБД MySQL версії не нижче 5.1 і веб-сервер з інтерпретатором PHP версії не нижче 5.2, для управління конфігураціями VM необхідний репозиторій Subversion. Для створення на сервері інфраструктури, необхідної для використання веб-інтерфейсу, потрібно створити БД на основі доданого до архіву з системою сценарію SQL. Також необхідно вказати необхідні параметри авторизації серверу MySQL і сховища Subversion в файлі **admin / config.inc.php** .

Робота служби спостереження за станом вузлів налаштовується за допомогою вказівки параметрів в XML-файл конфігурації служби, що знаходиться в одному каталозі з її JAR-файлом.

Всі налаштування служб агента також виконуються шляхом установки параметрів в XML-файлі конфігурації. Необхідно звернути увагу на ім'я вузла і пароль, які використовуються при авторизації оновлень і інших операцій - при неправильній установці цих параметрів вузол не зможе підтримувати комунікацію з іншими вузлами системи.

Для здійснення розгортання нових екземплярів VM на основі шаблону необхідно відкрити для загального доступу каталоги з файлом конфігурації і віртуальними жорсткими дисками шаблону VM. Щоб ці каталоги були доступні всім агентам, потрібно вказати відповідні параметри в їх файлах конфігурації.

Для здійснення міграції існуючих VM всі екземпляри Xen на всіх вузлах хмари повинні мати відповідні налаштування, а якщо при міграції планується використовувати SAN або інші спільні файлові сховища, то необхідно налаштувати і їх теж. Для отримання детальної інформації про налаштування Xen і файлових сховищ слід звернутись до відповідної документації цих продуктів.