ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ
Кафедра комп'ютерних інтелектуальних систем та мереж

Понятовська Наріне Самвелівна

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ВІРТУАЛЬНИХ МЕРЕЖ ДЛЯ ПОБУДОВИ ДИНАМІЧНИХ СИСТЕМ УЧБОВИХ ЛАБОРАТОРІЙ

Спеціалізація – Комп'ютерні системи та мережі
Спеціальності – 123 - Комп`ютерна інженерія

Керівник: Шапорін Р.О.
К.т.н., доцент

Одеса – 2021

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

Понятовська Наріне Самвелівна

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Дослідження технологій віртуальних мереж для побудови динамічних систем учбових лабораторій

керівник проекту (роботи) Шапорін Р.О. к.т.н, доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ректора ОНПУ від "_" 2020 року №

2. Строк подання студентом проекту (роботи) 01.12.2021

3. Вихідні дані до проекту (роботи) завдання на дослідження

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1 Дослідження технологій віртуальних віддалених мереж

2 Дослідження хмарних технологій для системи навчання

3 Моделі приватної учбової хмари віддаленої лабораторії

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Характеристика роботи, Модель мережевої структури віртуальної лабораторії, Модель окремого проекту, Модель комплексного проекту, Інформаційна модель створення учбового проекту, Структурна модель контролю віртуального проекту, Ієрархічна модель учбових даних, Модель зберігання даних, Переваги використання моделей, Висновки

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Дослідження технологій віртуальних віддалених мереж		
2	Дослідження хмарних технологій для системи		
	навчання		
3	Моделі приватної учбової хмари віддаленої лабораторії		
	Ілюстративний матеріал		

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ _____
(підпис) (прізвище та ініціали)

Відомість кваліфікаційної роботи магістра

№	Найменування	Кільк.	Примітка
1	Пояснювальна записка	70	
2	Характеристика роботи	1	
3	Модель мережевої структури віртуальної лабораторії	1	
4	Модель окремого проекту	1	
5	Модель комплексного проекту	1	
6	Інформаційна модель створення учбового проекту	1	
7	Структурна модель контролю віртуального проекту	1	
8	Ієрархічна модель учбових даних	1	
9	Модель зберігання даних	1	
10	Переваги використання моделей	1	
11	Висновки	1	
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			

				АМДП.ЗАМ151.0404			
Зм.	Лист	№ докум.	Підпис	Дата			
Розробив		Понятовська			Дослідження технологій віртуальних мереж для побудови динамічних систем учбових лабораторій	Літ.	Лист 1 / Листів 1
Перевірив		Шапорін РО					
Реценз.						«Одеська політехніка» ІКС КІСМ ЗАМ151	
Н. Контр.							
Затвердив							

АНОТАЦІЯ

Понятовська Н.С. Дослідження технологій віртуальних мереж для побудови динамічних систем учбових лабораторій — кваліфікаційна робота магістра. Одеса, 2021.: 70 стор., 10 рисунків., 5 таблиця., 9 джерел.

Об'єкт дослідження – процес організації приватних хмар зі змінною архітектурою.

Предмет дослідження – моделі та методи побудови динамічних комп'ютерних мереж.

Метою даної роботи є розробка моделей та алгоритмів побудови динамічних хмарних систем для учбових лабораторій з ціллю зменшення часу розгортання лабораторних робіт та оптимізації використання ресурсів за рахунок використання систем віртуалізації.

В роботі проведено дослідження існуючих розгортання учбових лабораторній з локальним та віддаленим доступом з ціллю виявлення особливостей. Також, розглянуто системи побудови хмарних систем та визначення їх придатності для використання в учбових системах.

В ході проведення дослідження було розроблено ряд моделей які описують організаційну, інформаційну та ієрархічну структуру приватної учбової хмарної системи з урахуванням повного циклу учбових проектів від розробки до винищення динамічної системи.

За результатами дослідження отримано технічну систему, яка дозволяє зберігати структури учбових проектів та розгортати їх виключно за вимогою, що дозволяє оптимізувати учбову діяльність студентів та економити апаратні ресурси інформаційних систем учбових закладів.

ХМАРНІ СИСТЕМИ, ВІДДАЛЕНИЙ ДОСТУП, КОМП'ЮТЕРНІ СИСТЕМИ, СИСТЕМИ ВІРТУАЛІЗАЦІЇ

ABSTRACT

Ponyatowska NS Research of virtual network technologies for construction of educational laboratories dynamic systems - master's thesis. Odessa, 2021 .: 70 pages, 10 drawings, 5 tables, 9 sources.

The object of study - the process of organizing private clouds with variable architecture.

The subject of research is models and methods of building dynamic computer networks.

The aim of this work is to develop models and algorithms for building dynamic cloud systems for training laboratories in order to reduce the time of deployment of laboratory work and optimize the use of resources through the use of virtualization systems.

The study of the existing deployment of educational laboratories with local and remote access in order to identify features. Also, systems for building cloud systems and determining their suitability for use in educational systems are considered.

During the study, a number of models were developed that describe the organizational, informational and hierarchical structure of the private educational cloud system, taking into account the full cycle of educational projects from development to destruction of the dynamic system.

According to the results of the research, a technical system was obtained, which allows to preserve the structures of educational projects and deploy them only on demand, which allows to optimize students' educational activities and save hardware resources of information systems of educational institutions.

CLOUD SYSTEMS, REMOTE ACCESS, COMPUTER SYSTEMS, VIRTUAL SYSTEMS

ЗМІСТ

ВСТУП

Актуальність. В умовах інтенсивної інформатизації учбового процесу перед викладачами, інженерами та адміністрацією учбових закладів постає складна задача використання ресурсів лабораторій в ході виконання учбових активності студентами.

В стандартному виконанні, студенти виконують завдання в лабораторія, які розраховані на 10-20 студентів для одночасної пристутності. Також, такий підхід передбачає роботу виключно за вказаним розкладом занять, що обмежує певні типи активності, наприклад дослідницькі роботи, розрахункові роботи та інше.

Метою даної роботи є розробка моделей та алгоритмів побудови динамічних хмарних систем для учбових лабораторій з ціллю зменшення часу розгортання лабораторних робіт та оптимізації використання ресурсів за рахунок використання систем віртуалізації.

Об'єкт дослідження – процес організації приватних хмар зі змінною архітектурою.

Предмет дослідження – моделі та методи побудови динамічних комп’ютерних мереж.

Методи дослідження. Основними методами дослідження були теорія графів, теорія множин, теорія нечітких мір, математична статистика.

В кваліфікаційній роботі магістра були розглянуті та досліджені різні технології. Ціль дослідження – виявлення найкращих технологій для досягнення мети. В першому розділі розглянуті принципи віртуальних мереж, віддаленого доступу до мереж та систем. Визначені основні технології та принципи використання їх. Виявлені основні переваги та недоліки, розглянуті архітектури та визначені найбільш захищені та

використовувані. Досліджено використання різноманітного програмного забезпечення по роботі з віддаленим доступом та захисту програмного забезпечення.

В другому розділі проводиться дослідження основних понять та визначень. Виявлення понять динамічних систем, понять віртуалізації та її типів, а також, хмарні сервіси. Визначаються основні характеристики хмарних обчислень та наведені їх класифікації. Дослідження проведено з ціллю визначити можливість використання технологій в учбовому процесі.

В ході дослідження моделей було запропоновано множину елементів, які дозволяють описувати динамічну систему учбової хмари, робота якої спрямована на економію обчислювальних ресурсів із зберіганням доступу студентів до учбового контенту в гнучкому стилі розкладу занять. Розроблені моделі вимагають реалізації у вигляді приватної хмари та віртуальної лабораторії.

1 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ВІРТУАЛЬНИХ ВІДДАЛЕНИХ МЕРЕЖ

У століття високих технологій та інформатизації суспільства інформація стає одним з найважливіших ресурсів бізнесу і головним елементом виробничих сил. У зв'язку з цим зростає попит і вартість на ексклюзивне право володіння інформацією. Тому все частіше ми стикаємося з тим, що право на володіння інформацією може бути порушено тим чи іншим способом, і бізнес зазнає колосальних збитків або взагалі припиняє свою діяльність.

Таким чином, одним з важливих аспектів діяльності будь-якого підприємства є захист інформаційних ресурсів компанії, тобто її інформаційна безпека, і диверсифікація ризиків пов'язаних з витоком інформації. [1]

Під поняттям «системи віддаленого управління» розуміються архітектури, протоколи, різноманітні програми і сценарії, які описують, що і коли потрібно робити для оптимального використання системи. Це стосується не тільки того, як розпізнати помилку і відреагувати на неї, а й повсякденних процесів, таких, як установка нового програмного забезпечення або додавання чергового користувача. Щоб все це здійснити децентралізовано і по можливості ефективно, потрібна відпрацьована технологія для організації віддаленого доступу до всіх функцій комп'ютера.

1.1 Технології віддаленого доступу

Технологія віддаленого доступу – програми або функції операційних систем, що дозволяють отримати віддалений доступ до комп'ютера через

Інтернет або локальну обчислювальну мережу (далі ЛОМ) і управління та адміністрування віддаленого комп'ютера в реальному часі.

Віддалений доступ надає майже повний контроль над віддаленим комп'ютером: він надає можливість дистанційно керувати робочим столом комп'ютера, можливість копіювання або видалення файлів, запуску додатків і так далі. [1]

Щоб правильно розставити акценти для різних технологій і зрозуміти їх, представлена відповідна класифікація (рисунок 1.1).

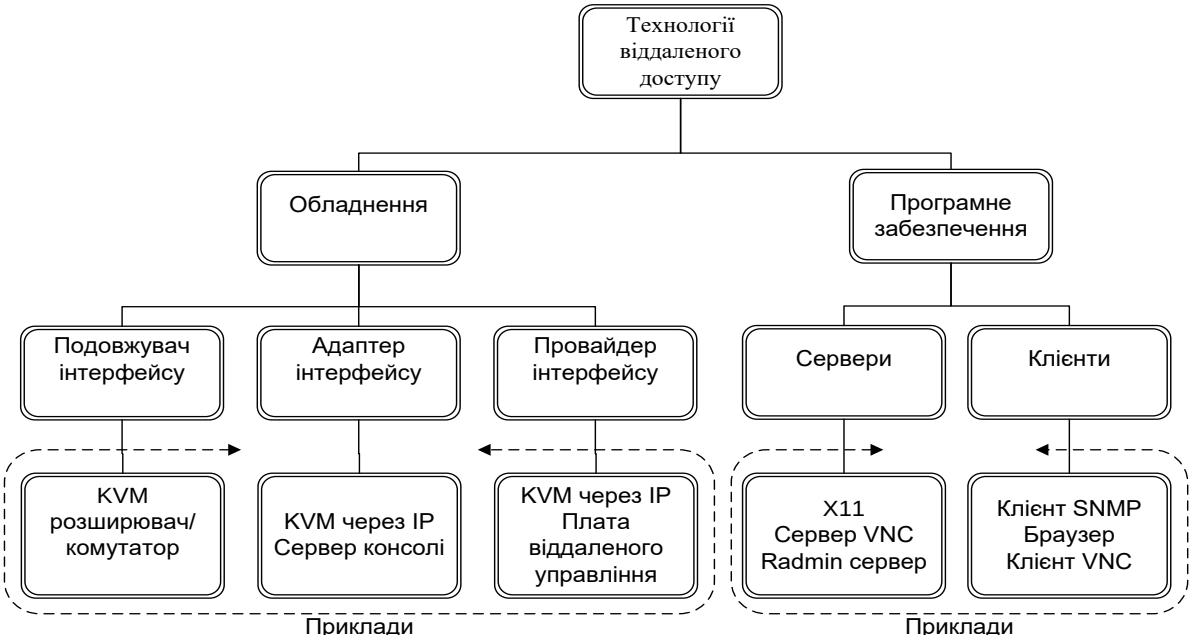


Рисунок 1.1 – Класифікація технологій віддаленого доступу

На рисунку 1.1 показано як застосовується класифікація різних технологій віддаленого управління. Протоколи і технології доступу мають принципові відмінності. Протоколи визначають, як структуруються пов'язані з управлінням дані і здійснюється обмін між ними. Вони дозволяють використовувати стандартизоване програмне забезпечення управління (клас: доступ / програмне забезпечення / клієнт), проте не специфікують, яким чином відповідні дані фізично видобуваються і обробляються. Це завдання технологій доступу, саме вони дозволяють дистанційно управляти пристроєм і віддалено запитувати його параметри. Принципово таке стає можливим за

рахунок додаткових апаратних засобів або програмного забезпечення, яке адміністратор встановлює на контрольованому пристрої. Додаткове апаратне забезпечення дозволяє «подовжити» існуючий інтерфейс, так що в цьому випадку говорять про «розширник». Крім того, розповсюдженим засобом є перетворення інтерфейсів в інші інтерфейси, такий апаратний засіб називають «адаптер». І нарешті, пристрій розширюють за допомогою інтерфейсу, який не перебуває в його розпорядженні, внаслідок чого такий вид апаратного забезпечення отримав назву «провайдер». Оскільки ми будемо розглядати можливості глобального управління, то перш за все мова піде про адаптери та провайдери з підтримкою доступу по TCP/IP. [1]

У разі сервера програмні технології – це програми, які використовують комунікаційні інтерфейси контрольованого пристрою і таким чином пропонують функціональність управління. У разі клієнтів - це призначені для користувача програми, взаємодія яких здійснюється за допомогою протоколу з будь-якою технологією доступу, програмної або апаратної, за умови, що вони спілкуються за допомогою того ж самого протоколу.

1.2 Критерії оцінювання технологій віддаленого управління

Деякі з наведених нижче критеріїв можуть допомогти при оцінці різних технологій. Однак через різницю між окремими класами не всі вони можуть бути застосовані для кожного випадку, а в основному розраховані на класи технологій доступу. Критерії впорядковані за ступенем важливості [2]:

– Незалежність від операційної системи. Виконання цієї ознаки означає, що, навіть якщо система не функціонує, комп'ютер все одно доступний і керований, що особливо важливо, якщо за допомогою рішення управління потрібно змінити базові конфігурації, наприклад настройки BIOS на віддаленому сервері.

– Зовнішні комунікаційні з'єднання. Ознака описує інтерфейсні рішення, через які можна діяти ззовні. Комунікації можуть здійснюватися по

зовнішньому (OutofBand, OoB) і основному (InBand, IB) каналам. При цьому можливість доступу по зовнішньому каналу виявляється корисна, якщо комунікаційні інтерфейси комп'ютера або мережі не працюють і вимагають переконфігурації, оскільки з'єднання по основному каналу було перервано так само, як і з'єднання з очікуючою системою.

– Доступ до коду завантаження / BIOS. В якості коду завантаження комп'ютера зазвичай позначають ту частину програмного забезпечення незалежної пам'яті, виконання якої відбувається безпосередньо після включення системи. З його допомогою задаються основні параметри системи і здійснюється управління самим процесом завантаження. Якщо код завантаження робочої станції зі скороченим набором команд (RISC) звертається найчастіше до послідовної консолі, то код завантаження комп'ютера (традиційно позначається як BIOS) передбачає наявність відповідної графічної карти VGA. Все це задовільно функціонує локально, віддалений же доступ до консолей VGA представляє складну задачу. З іншого боку, для повного доступу необхідна можливість зміни опцій коду завантаження.

– Безпека. Мета віддаленого управління полягає в забезпеченні глобального доступу до пристрою. Це також означає, що останнім зі своїми важливими і, в більшості випадків, уразливими інтерфейсами виявляється доступним для всього світу. Таким чином, ясно, що концепція безпеки рішення віддаленого управління повинна бути ретельно продумана і відпрацьована. Шифрування, підтвердження справжності і контроль доступу.

– Внутрішнє з'єднання. Критерій визначає, через які інтерфейси адміністратор здійснює віддалене управління контрольованим комп'ютером. Крім типових апаратних інтерфейсів, таких, як вихід VGA або кнопка скидання, поширені і програмні інтерфейси.

– Віртуальні пристрої. Вони дають можливість системі управління емулювати на комп'ютері обладнання, яке в дійсності до нього не приєднано. Найчастіше віртуальні пристрої використовуються для того, щоб

завантажити комп'ютер з альтернативного способу - диска, передача якого здійснюється через керуюче з'єднання.

− Програмне забезпечення управління. Призначене для обслуговування системи управління, воно в кінцевому підсумку визначає функціональність всього рішення. Найбільшого поширення набули дві концепції: система управління може використовувати нестандартний або стандартний протокол. У першому випадку потрібно застосування спеціального програмного забезпечення. У другому - можна застосовувати будь-який, що відповідає стандарту програмного забезпечення. Проміжне становище займає вибір в якості комунікаційного протоколу HTTP або telnet. Ці протоколи стандартизовані, отже, цілком підійде стандартне ПЗ, наприклад, браузер Web. Правда, вони не підтримують визначення мети та обміну інформації, яка описує функціональність рішень. Це означає, що власні інтерфейси користувачів, в свою чергу, нестандартні.

− Поширення. Поширеність рішення залежить від того, наскільки воно значуще практично або як перевірено фактично.

1.2.1 Огляд технологій віддаленого доступу згідно з запропонованими критеріями оцінювання

Далі детально розглядаються технології віддаленого доступу. Кожен розділ супроводжується таблицею, де система оцінюється відповідно до запропонованих критеріїв. Якщо критерій не застосовується, тоді він не наводиться.

Інтерфейс управління інтелектуальною платформою (IntelligentPlatformManagementInterface, IPMI) – IPMI це ініціатива різних виробників комп'ютерів. IPMI визначає низькорівневий інтерфейс і таким чином дозволяє контролювати фізичне функціонування хоста. Сюди відносяться різні параметри: температура, напруга, частота обертання вентилятора, блоки живлення, записи в журналі реєстрації подій і ін. Додатково IPMI включає механізми оповіщення про критичні стани системи,

автоматичного рестарту, дистанційного включення і відключення, а також скидання (таблиця 1).

Починаючи з версії 1.5 IPMI реалізується у вигляді так званого контролера управління платою (BaseboardManagementController, BMC) на системній платі хосту. BMC також здійснює доступ до налаштувань BIOS, який для ПК традиційно можливий тільки через консоль VGA.

Для ПК IPMI пропонує цікавий спосіб реалізації уніфікованого доступу до функцій хосту, здійснити який до цих пір вдавалося тільки з великими технічними труднощами. Оцінка IPMI системи представлена у таблиці 1.1.

Таблиця 1.1 – Оцінка системи віддаленого доступу «IPMI»

Клас	Доступ / обладнання / провайдер
Незалежність від ОС	Так
Зовнішній інтерфейс	IPMB (зовнішній канал) починає з версії 1.5, а також послідовний (зовнішній канал) Ethernet (мережа)
Доступ до BIOS	Так
Безпека	Аутентифікація
Внутрішній інтерфейс	Живлення, скидання, напруга, температура, вентилятор, журнал реєстрації несправностей
Віртуальні пристрої	Нема
ПЗ для клієнта	Нестандартне, SNMP
Застосування	Використовується на серверах Intel

Наступною технологією віддаленого доступу було розглянуто пристрої віддаленого управління. Пристрої віддаленого управління – це знімні карти PCI, головним чином для ПК. Як і IPMI, вони оснащують хост розширеними

і незалежними від операційної системи інтерфейсами для реалізації віддаленого доступу до комп'ютера.

Найважливіша властивість плат дистанційного керування – їх здатність керувати локальною консоллю. Ця обставина є вирішальною, тому що для більшості ПК консоль VGA представляє єдину можливість отримати доступ до BIOS і змінити послідовність завантаження таким чином, щоб операційна система запускалася з системи відновлення. Багато плат пропонують управління консоллю в графічному режимі, щоб забезпечити постійний доступ до хосту у всіх режимах роботи. Завдяки їх впровадженню в хост – систему виявляється можливим створювати віртуальні пристрої та віддалено передавати і завантажувати образ диска.

Плати віддаленого управління мають свої зовнішні комунікаційні інтерфейси і пропонують додатково також ще одну або кілька альтернатив живлення. Таким чином забезпечується незалежний доступ. У всіх відомих випадках вони виконані як автономний комп'ютер і тому досить дорогі. Велика частина функціональності подібної плати може бути реалізована і іншими засобами, але набагато складніше. З усіх представлених систем вони здійснюють найповніший і «найприродніший» віддалений доступ до комп'ютера за допомогою простого рішення «увімкни і працюй» [3].

У цьому причина їх надзвичайної популярності в управлінні комп'ютерними системами на базі ПК. На жаль, більшість відомих плат дистанційного керування залежні від виробника, тобто функціонують тільки в комп'ютерних системах які випускаються конкретною компанією і тільки в певних моделях. Лише плата дистанційного керування від Peppercon підходить практично для всіх і сама дбає про сумісність в частині роз’ємів скидання і живлення. Оцінка технології представлена у таблиці 1.2.

Таблиця 1.2 – Оцінка системи «пристрої віддаленого управління»

Клас	Доступ / обладнання / провайдер
Незалежність від ОС	Так
Зовнішній інтерфейс	Ethernet, модем, (ISDN) все по зовнішньому каналу
Доступ до BIOS	Так за допомогою управління з консолі
Безпека	SSL та авторизація
Внутрішній інтерфейс	Живлення, скидання, напруга, VGA(текст графіка), клавіатура, мишка, IPMI
Віртуальні пристрої	Так
ПЗ для клієнта	На базі WEB чи нестандартне
Застосування	Широко використовується в серверах частіше всього залежать від виробника.
Приклади	Eric (Peppercon), RemoteInsightLightsOut (HP)

Технології послідовної консолі (оцінка системи наведена в таблиці 1.3) часто застосовуються в якості універсальних інтерфейсів управління і традиційно підтримуються серверами RISC, комутаторами, маршрутизаторами і джерелами безперебійного живлення. Їх кардинальна перевага полягає в технічній простоті, значному поширенні і наявності давніх традицій, а тому вони представляють собою універсальний інтерфейс для самих різних пристроїв. Сервери консолей призначені для об'єднання доступу до декількох послідовних консолей за допомогою TCP / IP, так що, з одного боку, доступ до безлічі серверів стає можливим через одну-єдину точку, а з іншого - зазначений доступ виявляється глобальним.

Клієнтське ПЗ серверів консолі – це стандартні програми telnet або SSH. В принципі, сервер консолі можна реалізувати за допомогою звичайного комп'ютера і Linux, однак професійні пристрої відрізняються дуже високою щільністю портів і низькою ціною.

Таблиця 1.3 – Оцінка системи віддаленого доступу «Сервер консолі»

Клас	Доступ / обладнання / адаптер
Незалежність від ОС	Так
Зовнішній інтерфейс	Ethernet, модем, (ISDN) все по зовнішньому каналу
Доступ до BIOS	Так у ПК в яких є підтримка системної плати IPMI
Безпека	SSL
Внутрішній інтерфейс	RS232
Віртуальні пристрої	Нема
ПЗ для клієнта	Telnet, SSH
Застосування	В розрахункових центрах із комп’ютерами UNIX

Наприклад, в даний час пропонується сервер консолі висотою 1U з 48 послідовними інтерфейсами RS-232. Він призначений для обчислювальних центрів і провайдерів послуг, які хотіли б з меншими витратами і уніфіковано контролювати сотні серверів без графічного зовнішнього інтерфейсу і інші пристрої – від маршрутизатора до кондиціонера.

Спеціалізоване програмне забезпечення віддаленого доступу також є однією із технологій яка дозволяє дистанційно керувати комп'ютером. Для цього необхідно лише, щоб був встановлений серверний компонент ПЗ і хост мав з'єднання з середовищем передачі. Прикладом може служити вільне програмне забезпечення таке як віддалений робочий стіл (RDP), VNC, Radmin, TeamViewer, AmmyyAdmin і т. д., які для всіх популярних систем і архітектур пропонують як серверні, так і клієнтські версії. Програмні рішення цілком задовольняють багатьом з наведених критеріїв: зокрема, вони без проблем реалізуються по зовнішньому каналу, а в частині передачі графічного змісту монітора вони перевершують свої апаратні аналоги,

оскільки можуть скористатися істотно більш повною і точною інформацією для ефективного кодування зображення. Для них характерно неперевершене співвідношення продуктивності і ціни. Для оцінки віддаленого доступу через програмне забезпечення розглянемо таблицю 1.4:

Таблиця 1.4 - Оцінка системи віддаленого доступу через ПЗ

Клас	Протокол
Безпека	TLS, Аунтентифікація, авторизація, шифрування, (все залежить від ПЗ)
ПЗ для клієнта	Будь яке ПЗ яке підтримує SNMP
Застосування	Широко використовується в мережевому обладнані та мережевому управлінні

1.2.2 Дослідження програмного забезпечення віддаленого доступу

Програми забезпечення віддаленого доступу - це програми або функції операційних систем, що дозволяють отримати віддалений доступ до комп'ютера через Інтернет або локальну обчислювальну мережу (далі ЛОМ) і управління та адміністрування віддаленого комп'ютера в реальному часі. Віддалений доступ надає майже повний контроль над віддаленим комп'ютером: програми дають можливість дистанційно керувати робочим столом комп'ютера, можливість копіювання або видалення файлів, запуску додатків і так далі. Проте при виході з ладу операційної системи рішення на базі програмного забезпечення безсиле.

Далі розглянуто програмне забезпечення віддаленого доступу до інформаційних систем для кращого розуміння роботи програмного забезпечення.

Першою розглянемо Virtual Network Computing (VNC) - система віддаленого доступу до робочого столу комп'ютера, що використовує протокол RFB (англ. Remote Frame Buffer, віддалений кадровий буфер).

Управління здійснюється шляхом передачі натискань клавіш на клавіатурі і рухів миші з одного комп'ютера на інший і ретрансляції вмісту екрану через комп'ютерну мережу.

Система VNC - кросплатформне програмне забезпечення, яке не залежить від вибору ОС сімейств Windows чи UNIX\Linux: VNC-клієнт, званий VNC viewer, запущений на одній операційній системі, може підключатися до VNC-сервера, що працює на будь-який інший ОС. Існують реалізації клієнтської і серверної частини практично для всіх операційних систем, в тому числі і для Java (включаючи мобільну платформу J2ME). До одного VNC-сервера одночасно можуть підключатися декілька клієнтів. Найбільш популярні способи використання VNC - віддалена технічна підтримка і доступ до робочого комп'ютера з віддаленого робочого місця.

VNC складається з двох частин: клієнта і сервера. Сервер - програма, що надає доступ до керування периферійними пристроями комп'ютера, а також до графічного інтерфейсу комп’ютера, на якому вона запущена. Клієнт (або viewer) - програма, яка отримує доступ до керування периферійними пристроями комп'ютера, а також отримує зображення екрану з сервера і взаємодіє з ним по протоколу RFB.

Програмне забезпечення RFB (англ. Remote frame buffer) - простий клієнт-серверний мережевий протокол прикладного рівня для віддаленого доступу до графічного робочого столу комп'ютера, який використовується в VNC. Так як він працює на рівні кадрового буфера, то його можна застосовувати для графічних віконних систем, наприклад, X Window System, Windows, Quartz Compositor.

За замовчуванням RFB використовує діапазон TCP-портів з 5900 до 5906. Кожен порт є відповідний екран X-сервера (порти з 5900 по 5906 асоційовані з екранами з 0 по 6). Java-клієнти, доступні в багатьох реалізаціях, що використовують вбудований веб-сервер для цієї мети, наприклад, в RealVNC, пов'язані з екранами таким же чином, але на діапазоні портів з 5800 до 5806. Багато комп'ютерів під управлінням ОС Windows

можуть використовувати лише один порт через відсутність багатокористувацьких властивостей, властивих UNIX-системам. Для Windows-систем екран за замовчуванням - 0, що відповідає порту 5900.

Також існує можливість зворотного підключення від сервера до клієнта. У цьому випадку клієнт переводиться в режим прослуховування і з'єднання ініціюється сервером на 5500 TCP-порт клієнта. Порти можуть бути змінені.

Спочатку VNC не використовує шифрування трафіку, проте в процедурі аутентифікації пароль не передається у відкритому вигляді, а використовується алгоритм «виклик-відповідь» з DES-шифруванням (ефективна довжина ключа становить 56 біт). У багатьох реалізаціях не може перевищувати 8 символів на довжину пароля і якщо його довжина перевищує 8 символів, то пароль урізається, а зайві символи ігноруються.

При необхідності надійного шифрування всієї VNC-сесії, вона може бути встановлена через SSL, SSH або VPN-тунель, а також поверх IPsec. Технологія IPsec підтримується переважною більшістю сучасних ОС і використовується як при з'єднанні через Інтернет, так і в локальних мережах. SSH-клієнти дозволяють створювати SSH-тунелі як для всіх основних платформ (Linux, BSD, Windows, Macintosh і ін.), так і для менш популярних.

Також багато сучасних версій VNC підтримують розширення стандартного протоколу, які реалізують шифрування і / або стиснення VNC-трафіку, розмежування за списками доступу ACL і різні методи аутентифікації.

EchoVNC використовує OpenSSL для шифрування з'єднань, причому шифрується сесія VNC, включаючи аутентифікацію і передачу даних. Також підтримує передачу файлів і чат. Якщо клієнт не підтримує OpenSSL шифрування, то шифрування автоматично відключається.

UltraVNC дозволяє використовувати спеціальний плагін, який поширюється з відкритим вихідним кодом, який шифрує всю сесію VNC використовуючи алгоритми AES або RC4, включаючи аутентифікацію і

передачу даних. Також існують варіанти аутентифікації на основі NTLM і облікових записів користувачів в ActiveDirectory. UltraVNC дозволяє передавати файли між сервером і клієнтом в будь-яких напрямках.

RealVNC в комерційній версії продукту використовує алгоритм AES для шифрування з'єднання і алгоритм RSA для аутентифікації. Workspot випустила патч для VNC, який реалізує алгоритм шифрування AES.

PuTTY – вільно розповсюджуваний клієнт для різних протоколів віддаленого доступу, включаючи SSH, Telnet, rlogin. Також є можливість роботи через послідовний порт. PuTTY дозволяє підключитися і керувати віддаленим вузлом (наприклад, сервером). У PuTTY реалізована тільки клієнтська сторона сполуки - сторона відображення, в той час як сама робота виконується на стороні сервера. Спочатку розроблявся для Microsoft Windows, проте пізніше портований на Unix. У розробці знаходяться порти для Mac OS і Mac OS X. Сторонні розробники випустили неофіційні порти на інші платформи: мобільні телефони під управлінням Symbian OS, комунікатори з Windows Mobile, а також пристрої з iOS і Android.

PuTTY входить в репозиторії практично всіх основних систем Linux (в т.ч. Ubuntu, Debian, ALT Linux). Вихідний код PuTTY повністю розроблений на C. PuTTY не залежить від DLL, інших додатків, пакетів оновлень ОС. Пакет складається тільки з виконуваних файлів, які можуть бути встановлені в будь-якому місці. PuTTY і більшість утиліт запускаються тільки в одному потоці ОС. Програма є вільно поширюваним додатком з відкритим вихідним кодом і випускається під OpenSource ліцензією.

До можливостей програми можемо віднести роботу з ключами і версіями протоколу SSH, клієнтами SCP та SFTP (відповідно програми pscp і psftp); можливість перенаправлення портів через SSH, включаючи передачу X11; підтримка більшої частини керівних послідовностей xterm, VT-102, а також значна емуляція терміналу ECMA-48;підтримка 3DES, AES, Arcfour, Blowfish, DES; підтримка аутентифікації з відкритим ключем, в тому числі і

без введення пароля; підтримка роботи через послідовний порт (починаючи з версії 0.59); можливість роботи через проксі – сервер.

Застосування PuTTY можемо знайти в наступних завданнях: віддалене адміністрування Linux, підключення до віртуальних серверів по протоколу SSH, налаштування мережевих маршрутизаторів через послідовний порт, з'єднання з віддаленими Telnet-терміналами.

AnyDesk -це програма для організації віддалених сесій, що дозволяє отримати доступ до дистанційного комп'ютера так, як ніби ви фізично перебуваєте за ним. Програма застосовує спеціальний алгоритм обробки картинки, при якому передаються тільки частини зображення, які змінилися. Таким чином, трансляція залишається максимально чіткою навіть при неякісному Інтернет - з'єднанні. Для комерційних користувачів передбачена функція неконтрольованого доступу. З її допомогою можна приєднуватися до машини без підтвердження, також є можливість блокування пристроїв введення (клавіатура, миша). Головна відмінність, яку відзначають розробники – це висока швидкість роботи AnyDesk в порівнянні з усіма іншими аналогічними програмами.

TeamViewer– пакет програмного забезпечення для віддаленого контролю комп'ютерів спільного використання, обміну файлами між керуючою і керованою машинами, відеозв’язку та веб-конференцій. TeamViewer працює на операційних системах Microsoft Windows, Mac OS X, Linux, Chrome OS, iOS, Android, RT Windows, BlackBerry і Windows Phone 8.Крім прямого з'єднання, доступ можливий через брандмауер і NAT проксі, можливо отримання доступу до віддаленої машини за допомогою веб-браузера. TeamViewer може використовуватися безкоштовно некомерційними користувачами. Корпоративні версії також доступні.

Встановлення зв'язку. TeamViewer може працювати з установкою або без неї - в останньому випадку програма працює без адміністраторських прав доступу. Для встановлення зв'язку TeamViewer повинен бути запущений на обох машинах. При запуску TeamViewer створюється ID комп'ютера і пароль.

Щоб встановити зв'язок між комп'ютерами, клієнт-оператор повинен зв'язатися з віддаленим оператором і дізнатися його ID і пароль, а потім ввести їх в клієнт-TeamViewer.

TeamViewer також може встановити зв'язок з віддаленим комп'ютером, використовуючи браузер з технологією Flash. У конфігурації за замовчуванням TeamViewer використовує один з серверів TeamViewer.com, щоб запустити з'єднання і маршрутизацію трафіку між локальним клієнтом і віддаленою хост-машиною. Програмне забезпечення тоді визначає, як встановити з'єднання. У 70% випадків після підтвердження з'єднання встановлюється пряме підключення через UDP або TCP; інші сполуки спрямовані через мережу маршрутизатора TeamViewerGmbH (через TCP або тунелювання HTTP).

TeamViewer дозволяє встановлювати VPN-з'єднання (VirtualPrivateNetwork) між клієнтом і сервером. Є можливість завантажити з сайту виробника окремі модулі програми (клієнтський і серверний). Можна також на сайті виробника конфігурувати клієнтський модуль з заздалегідь встановленим паролем доступу і власним логотипом, скомпілювати і відразу завантажити його. Однак, без ліцензії, зв'язок в цьому випадку можливий не більше п'яти хвилин за сеанс. Пропоновані модулі без власних попередніх даних не мають таких обмежень. Ці модулі не вимагають інсталяції і прості у використанні. Можливий відео, голосовий, і текстовий чат між комп'ютерами.

TeamViewer забезпечує повне шифрування на основі обміну особистими/ публічними ключами RSA 2048 та шифрування сеансів AES (256-бітному). Ця технологія заснована на тих самих стандартах, як і https /SSL, і відповідає сучасним стандартам захисту даних. Обмін ключами також гарантує повну безпеку даних, що передаються від клієнта до клієнта. Це означає, що наші сервери-маршрутизатори не можуть зчитати потік даних. Всі програмні файли захищені за допомогою технології підпису коду VeriSign. Це дозволяє перевірити джерело отриманих вами файлів.

Із метою забезпечення додаткового захисту від несанкціонованого доступу до системи на додаток до ID партнера TeamViewer генерує також пароль сеансу, що змінюється при кожному запуску програмного забезпечення. Окремі функції, що впливають на безпеку, зокрема передача файлів, вимагають додаткового підтвердження вручну від віддаленого партнера. Також неможливо непомітно керувати комп'ютером. Із метою захисту даних особа, яка сидить за віддаленим комп'ютером, повинна мати можливість визначити, коли хтось намагається отримати доступ до машини.

TeamViewer допомагає компаніям у виконанні вимог відповідності HIPAA і PCI. Дворівнева аутентифікація робить внесок у підвищення рівня безпеки для захисту облікових записів TeamViewer від несанкціонованого доступу. У поєднанні з контролем доступу через рекомендований список TeamViewer допомагає використовувати HIPAA і PCI. Із дворівневою аутентифікацією для входу в обліковий запис TeamViewer додатково до імені користувача та паролю потрібен код, згенерований на мобільному пристрої. Код генерується на основі алгоритму одноразового тимчасового паролю (TOTP). Код TOTP захищений за допомогою SRP і, таким чином, має прекрасну стійкість проти комп'ютерних атак зловмисників.

Програма Radmin призначена для віддаленого контролю за TCP / IP для Windows. Поряд з TeamViewer користується популярністю в отриманні віддаленого управління інформаційними системами.

Додатки Radmin Server і RadminViewer встановлюються, відповідно, на віддаленій і локальній машинах. На сайті розробника можна завантажити кожен з модулів окремо або одним архівом, є портативна версія і пакет для мережевої установки. Після майстра установки і автоматичної інсталяції необхідних драйверів потрібне налаштування серверної частини - користувач вказує режим запуску, права доступу та інші параметри. Налаштовувати брандмауер немає необхідності, в разі зайнятості порту можна вказати будь-який вільний.

Підтримуються наступні режими роботи: віддалене управління комп'ютером, перегляд, telnet, передача файлів, вимикання, текстовий і голосовий чати.

При з'єднанні з віддаленим комп'ютером необхідно знати його IP: в Radmin не використовується Інтернет-ID і пароль. Це може здатися незручним, і розробники з цього приводу дають пораду: «використовувати програми сторонніх розробників, які дозволяють відслідковувати зміну IP адреси і замінюють його постійним DNS адресою». Крім того, у випадку з віртуальним з'єднанням також потрібно використовувати сторонні методи.

Слід сказати, що чергова перевага клієнта Radmin примітна тим, що під час налаштування пропонує вибрати метод установки прав доступу - Radmin або Windows NT. Різниця полягає в наступному: перший режим оптимальний для з'єднання по Інтернету, а методи Windows більш виправдані для корпоративних мереж. До того ж, права доступу Radmin зручні своїм вибірковим налаштуванням доступу до важливих функцій, плюс, можна скласти список довірених користувачів.

Всі передані дані шифруються за стандартом AES, з випадковою генерацією ключа. Паролі не передаються через сервер, замість цього Radmin використовує хешування (контрольну суму). Передбачений інтелектуальний захист від підбору пароля. Вхідні підключення до сервера можливі за запитом, щоб уникнути несанкціонованого доступу.

Простий протокол керування мережею (SimpleNetworkManagementProtocol, SNMP), на відміну від вже згаданих технологій, використовується для обміну інформацією, що управляє і є одним з найвідоміших стандартів «де-факто». Специфікація SNMP представлена в третій версії і складається з наступних пунктів:

− формальна мова для визначення даних (абстрактна нотація синтаксису версії 1, мова ASN.1 (Abstract Syntax Notation, ASN.1));

− визначення керуючої інформації (база керуючої інформації (Management Information Base, MIB));

− визначення протоколу;

− визначення безпеки і адміністрування.

SNMP передбачає розподілену архітектуру управління з адміністраторами та агентами. Менеджер запитує дані у агента відповідно до моделі клієнт-сервер, оцінює їх і при необхідності змінює. Агент в свою чергу теж може послати дані менеджеру за допомогою так званих переривань. Модель є дуже гнучко розширюється завдяки формальній мові і MIB. Протокол підкуповує своєю універсальністю та простотою. Виробники мережевих компонентів роблять ставку на SNMP як спосіб контролю за своїми продуктами. Багато з названих систем віддаленого управління мають інтерфейс SNMP. Один з істотних недоліків полягає в тому, що захищена третя версія досі слабо поширена.

Управління підприємством на базі Web (WebBasedEnterpriseManagement, WBEM) - наступний незалежний від виробника протокол для уніфікованого контролю і конфігурації систем, мереж і додатків. WBEM специфікується робочою групою з розподіленого управління (DistributedManagementTaskForce, DMTF), в її роботі беруть участь багато відомих виробників.

Основу WBEM становить загальний інформаційний протокол (Common Information Protocol, CIM), метод формального структурування інформації, що управляє. Таким чином, CIM - прямий аналог бази керуючої інформації SNMP. CIM використовує об'єктно-орієнтовану конструкцію для опису схеми управління. WBEM визначає відповідність між XML і CIM, а отже, форму, в якій дані CIM можуть бути збережені в документах XML для зручного обміну через Web. Відповідний протокол Web описується таким же чином і може розглядатися як попередник популярного простого протоколу доступу до об'єкта (SimpleObject Access Protocol, SOAP).

У тісному зв'язку з WBEM знаходиться ініціатива мережі з підтримкою каталогів (DirectoryEnabledNetwork), де описується відображення CIM на службу каталогів LDAP. WBEM реалізований, наприклад, в Microsoft

Windows і SunSolaris і дозволяє впровадити інструменталізацію, а значить, і керованість компонентами операційної системи і додатків. Застосування WBEM в пристроях доступу до апаратних засобів до цих пір зустрічається не часто.

1.2.3Дослідження технологій захисту програмного забезпечення для віддаленого доступу

Усі розглянуті вище програмні забезпечення та системи віддаленого доступу використовують криптографічний протокол, який надає можливість на більш безпечний зв'язок - це протоколи SSL/TSL. Протокол TLS (transportlayersecurity) заснований на протоколі SSL (SecureSocketsLayer), спочатку розробленому в Netscape для підвищення безпеки електронної комерції в Інтернеті. Протокол SSL був реалізований на application-рівні, безпосередньо над TCP (Transmission Control Protocol), що дозволяє більш високорівневих протоколів (таким як HTTP/HTTPS) працювати без змін. Якщо SSL налаштований коректно, то сторонній спостерігач може дізнатися лише параметри з'єднання (наприклад, тип використовуваного шифрування), а також частоту пересилання і приблизну кількість даних, але не може читати і змінювати їх.

Після того, як протокол SSL був стандартизований IETF (Internet Engineering Task Force), він був перейменований в TLS. Тому хоча імена SSL і TLS взаємозамінні, вони все-таки відрізняються, так як кожне описує іншу версію протоколу .

Протокол TLS призначений для надання трьох послуг всім додаткам, які працюють над ним, а саме: шифрування, аутентифікацію і цілісність. Технічно, не всі три можуть використовуватися, однак на практиці, для забезпечення безпеки, як правило використовуються всі три:

− Шифрування - приховування інформації, переданої від одного комп'ютера до іншого;

− Аутентифікація - перевірка авторства переданої інформації;

− Цілісність - виявлення підміни інформації підробкою.

Як вже говорилося, TLS був розроблений для роботи над TCP, однак для роботи з протоколами дейтаграм, такими як UDP (UserDatagramProtocol), була розроблена спеціальна версія TLS, що отримала назву DTLS (DatagramT ransport Layer Security).

Основні кроки процедури створення захищеного сеансу зв'язку розглянутих вище системах віддаленого доступу:

− клієнт підключається до сервера, що підтримує TLS, і запитує захищене з'єднання;

− клієнт надає список підтримуваних алгоритмів шифрування і хеш-функцій;

− сервер вибирає зі списку, наданого клієнтом, найбільш надійні алгоритми серед тих, які підтримуються сервером, і повідомляє про свій вибір клієнта;

− сервер відправляє клієнту цифровий сертифікат для власної аутентифікації. Зазвичай цифровий сертифікат містить ім'я сервера, ім'я засвідчувального центру сертифікації і відкритий ключ сервера;

− клієнт, до початку передачі даних, перевіряє валідність (автентичність) отриманого серверного сертифіката, щодо наявних у клієнта кореневих сертифікатів засвідчувальних центрів (центрів сертифікації). Клієнт також може перевірити відкликання серверного сертифікату, зв'язавшись з сервісом засвідчувального центру;

− для шифрування сесії використовується сеансовий ключ. Отримання загального секретного сеансового ключа клієнтом і сервером проводиться по протоколу Діффі-Хеллмана. Існує історичний метод передачі згенерованого клієнтом секрету на сервер, за допомогою шифрування асиметричною криптосистемою RSA (використовується ключ з сертифікату сервера). Даний метод не рекомендований, але іноді продовжує зустрічатися на практиці.

На цьому закінчується процедура підтвердження зв'язку. Між клієнтом і сервером встановлено безпечне з'єднання, дані, що передаються по ньому,

шифрують і розшифровуються за використанням симетричної криптосистеми до тих пір, поки з'єднання не буде завершено.

При виникненні проблем на деяких з вищевказаних кроків підтвердження зв'язку може завершитися з помилкою, а безпечне з'єднання не буде встановлено.

1.3 Дослідження методів віддаленого доступу до інформаційних джерел

На основі аналізу способів віддаленого доступу до інформаційних систем було виявлено декілька методів організації віддаленого доступу клієнта до інформаційних систем. В даних методах використовуються наступні об’єкти:

− Прямий віддалений доступ до інформаційних джерел за допомогою клієнта віддаленого робочого столу – програмне забезпечення кросплатформне і тому може бути реалізовано де завгодно, в тому числі на планшетах і смартфонах. Такий клієнт використовує свій протокол RDP (RemoteDesktopProtocol), винайдений компанією Citrix та куплений компанією Microsoft;

− VPN (Virtual Private Network - віртуальна приватна мережа*)* – узагальнена назва методу, що дозволяє забезпечити одне або кілька мережних з'єднань (логічну мережу) поверх іншої мережі (наприклад, Інтернет). Незважаючи на те, що комунікації здійснюються по мережах з меншим або невідомим рівнем довіри (наприклад, по публічних мереж), рівень довіри до побудованої логічної мережі не залежить від рівня довіри до базових мереж завдяки використанню засобів криптографії (шифрування, аутентифікації, інфраструктури відкритих ключів, засобів для захисту від повторів і змін переданих по логічній мережі повідомлень).

Залежно від застосовуваних протоколів і призначення, VPN може забезпечувати з'єднання трьох видів:

- вузол – вузол;
- мережу – мережу.

Причому огляд фокусується на описі тільки тих технологій, застосування яких забезпечує доступ до всіх функцій комп'ютера. З їх допомогою можна застосовувати високорівневі протоколи і функції для вирішення завдання. Адже для успішного адміністрування потрібна так само і інсталяція нової операційної системи, відключення і включення системи, налаштування опцій коду завантаження або BIOS і переконфігурація маршрутизаторів.

1.3.1Метод прямого віддаленого доступу

Перший метод який буде розглянуто – це прямий доступ до сервера за допомогою клієнта віддаленого робочого столу. З точки зору технічної реалізації, то для клієнта не потрібні ніякі додаткові ресурси і обладнання. Клієнт віддаленого робочого столу або Remote Deskop Client - програмне забезпечення крос-платформне і тому може бути реалізовано де завгодно, аж до планшетів і смартфонів. Такий клієнт використовує свій протокол RDP (Remote Desktop Protocol), куплений Microsoft у компанії Citrix. Детальніше на рисунку 1.2.

Рисунок 1.2 – Принцип роботи сервера терміналу

Безпека ж цього рішення не є найкращою. Використовуючи це рішення, піддається небезпеці не тільки передана інформація, але і відкривається серйозна вразливість в безпеці самого сервера, що не може відповідати вимогам адекватної концепції безпеки. Вразливість в безпеці сервера обумовлюється тим, що ми відкриваємо прямий доступ на сервер не тільки клієнтам, але і всім користувачам Інтернет і відповідно сервер піддається різним атакам ззовні, що дуже небезпечно, оскільки в історії протоколу RDP можна нарахувати кілька критичних аспектів, за допомогою яких зламували сервери.

І якщо безпеку переданої інформації на сервер за допомогою цієї технології можна зашифрувати вбудованими засобами клієнта віддаленого робочого столу (можливість використання 128-бітове шифрування по алгоритмам RC4, AES або 3DES з перевіркою цілісності хешем MD5 або SHA1, використання TLS шифрування), то вразливість в безпеки сервера при прямому доступі до сервера закрити неможливо. Далі розглянуті позитивні та негативні властивості такої технології.

Перевагами такого методу є:

− зручний і швидкий доступ з будь-якого місця з будь-якого пристрою без попереднього налаштування.

Недоліками є:

− вразливість в безпеці сервера;

− слабка захищеність переданої інформації.

1.3.2 Метод віртуальної приватної мережі

Наступний спосіб доступу до сервера ґрунтується на технології VPN, тобто узагальнена назва технологій, що дозволяють забезпечити одне або кілька мережевих з'єднань (логічну мережу) поверх іншої мережі (наприклад, інтернет).

Логічна мережа будується на основі програмного забезпечення OpenVPN. Це Open Source програмне забезпечення і може бути використано безкоштовно.

На практиці застосовується декілька типів з'єднань, використовуючи це програмне забезпечення:

− вузол-вузол;

− мережа-мережа.

Тунель типу вузол – вузол використовується у випадках, коли необхідно підключити одне робоче місце. Скажімо, наприклад користувачі територіально розкидані по всій країні, тоді логічніше буде використовувати саме такий тип з'єднання.

Для організації такого підключення необхідно буде встановити клієнт OpenVPN і провести попередні налаштування (описати конфігураційний файл - куди підключатися, який алгоритм шифрування використовувати, які сертифікати використовувати). Для підключення, користувачеві необхідно володіти спеціальним сертифікатом (точніше навіть декількома сертифікатами), за допомогою яких і буде виконувати шифрування інформації, при передачі її на сервер. Сертифікати виконують і ще одну важливу функцію - можливість ідентифікувати, хто саме підключився на сервер. Клієнт OpenVPN також є крос-платформним. Для забезпечення безпеки керуючого каналу і потоку даних, OpenVPN використовує бібліотеку OpenSSL. Завдяки цьому задається весь набір алгоритмів шифрування, доступних в даній бібліотеці.

Перевагою використання тунелю вузол – вузол може бути використання спеціального ключа для зберігання сертифікатів для підключення до сервера. Такий електронний ключ називають донгл. Зараз донгл частіше зустрічається з usb форм-фактором. Значення такого рішення полягає в тому, що на донгл записується і шифрується сертифікат користувача. Підключення до сервера здійснюється тільки за умови підключеного до пристрою донгла. Сертифікат на донгла шифрується і зазвичай захищений паролем, тому доступ на сервер виходить з 3-х факторним захистом:

− отримання фізичного доступу до донгла і підключення його до

пристрою;

− пароль до сертифікату на донгла для побудови тунелю між клієнтом і сервером;

− пароль для входу на сервер.

Використання технології зображеної на рисунку 3 вигідно в тому випадку, коли робоче місце користувача, який працює з віддаленим інформаційним джерелом, знаходячись в публічному місці з відкритим доступом.

Перевагами для такої технології являється:

− Крос-платформне рішення; використання особистих сертифікатів користувачів для безпеки інформації та ідентифікації користувача;

− використання шифрування трафіку між сервером і клієнтом;

− можливість використання апаратних ключів для доступу;

До недоліків такої системи можна віднести:

− Необхідність попереднього налаштування робочої станції користувача;

− підтримка робочої станції користувача ПЗ OpenVPN;

− попереднє налаштування кожного пристрою, з якого необхідно отримати доступ.

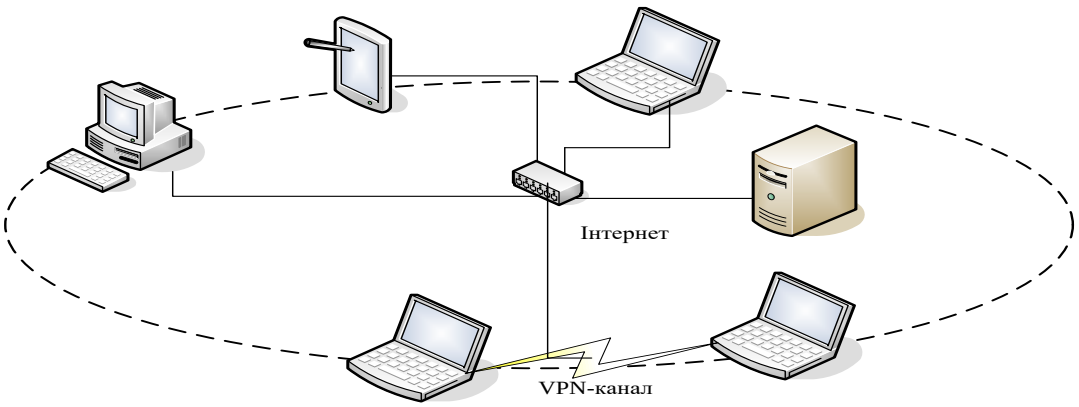


Рисунок 1.3 – VPN тунель точка – точка

Наступний тип з'єднання, використовуючи OpenVPN, є мережа-мережа. При використання такого типу VPN рекомендується використовувати невеликим підприємствам. Якраз таким підприємствам, яким немає сенсу купувати професійне мережеве обладнання з причини відсутності великого навантаження, можна придбати пристрої типу SOHO і доналаштувань до нього. Використовуючи спеціальне програмне забезпечення для такого типу пристроїв, можна домогтися його працездатності для побудови тунелю OpenVPN. Крім використання SOHO пристроїв, можна використовувати Linux маршрутизатори.

Прикладом таких SOHO пристроїв можуть послужити маршрутизатори Mikrotik. Таким чином, як і говорилося вище, такий тип підключення рекомендується невеликим компаніям з чисельністю співробітників в середньому до 15 осіб або менше, що знаходяться територіально в одній локально-обчислювальній мережі.

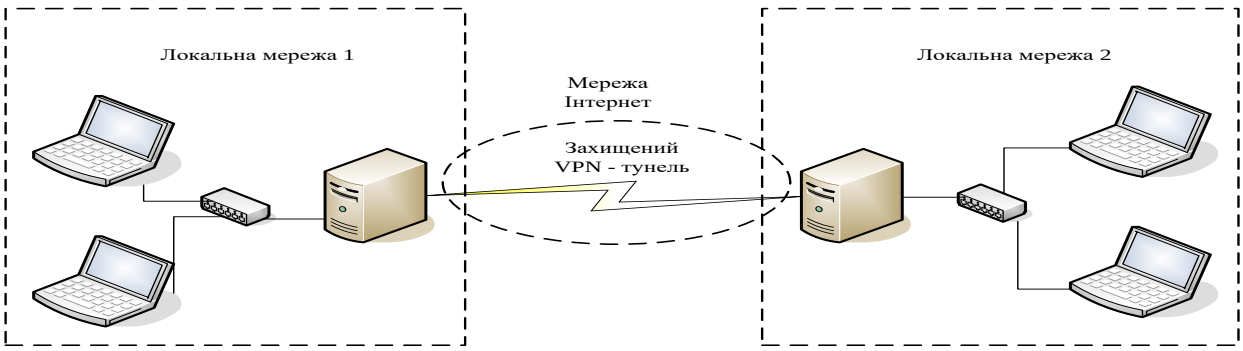


Рисунок 1.4 – VPN тунель мережа – мережа

Переваги цієї технології можна виразити в наступному:

− одна логічна мережа між сервером і офісом підприємства;

− потрібні попередні налаштування тільки одного пристрою, що відповідає за канал між сервером і користувачами;

− можливість використання на сервері ресурсів офісу замовника (мережеві принтера, файлові сховища, сховища резервних копій)

До недоліків можна віднести наступні проблеми:

− неможливість витримати серйозне навантаження;

− для шифрування і передачі інформації на стороні сервера використовується не спеціалізоване обладнання, а ресурси сервера, що значно зменшує швидкодію сервера.

Останнім з розглянутих типів з'єднання з серверів буде IPSEC.

Цей тип VPN з'єднання ми використовуємо для підключення VPN типу мережа-мережа. Для організації такого тунелю ми використовуємо професійне мережеве обладнання таких вендорів як Cisco, Juniper. Під час налаштування тунелів ми можемо враховувати наступну інформацію:

− Симетричні алгоритми для шифрування / розшифрування даних.

− Криптографічні контрольні суми для перевірки цілісності даних.

− Спосіб ідентифікації вузла. Найпоширеніші способи - це попередньо встановлені ключі (pre-sharedsecrets) або RSA сертифікати.

− Чи використовувати режим тунелю або режим транспорту.

− Яку використовувати групу DiffieHellman.

− Як часто проводити переідентіфікацію вузла.

− Як часто міняти ключ для шифрування даних.

Все це дозволяє нам встановити потрібний рівень шифрування, а також балансувати між рівнем шифрування і швидкодією маршрутизатора. Відповідно таке рішення ми можемо рекомендувати як великим, так і маленьким підприємствам з високим і не дуже високим рівнем передачі даних.

Використання професійного обладнання дозволяє нам організувати роботу користувачів навіть не дивлячись на можливі обмеження провайдерів і корпоративних фаєрволів. Прикладом такої технології може бути Cisco SSL VPN, який забезпечує шифрування з'єднання типу точка-мережа і використовує HTTPS (порт 443) для пересилання трафіку між клієнтом і сервером. Така технологія також називається WebVPN. Логін і аутентифікація кінцевого користувача здійснюється за допомогою веб-

браузера за допомогою запиту HTTP. Цей процес створює сесію, на яку посилається Cookies.

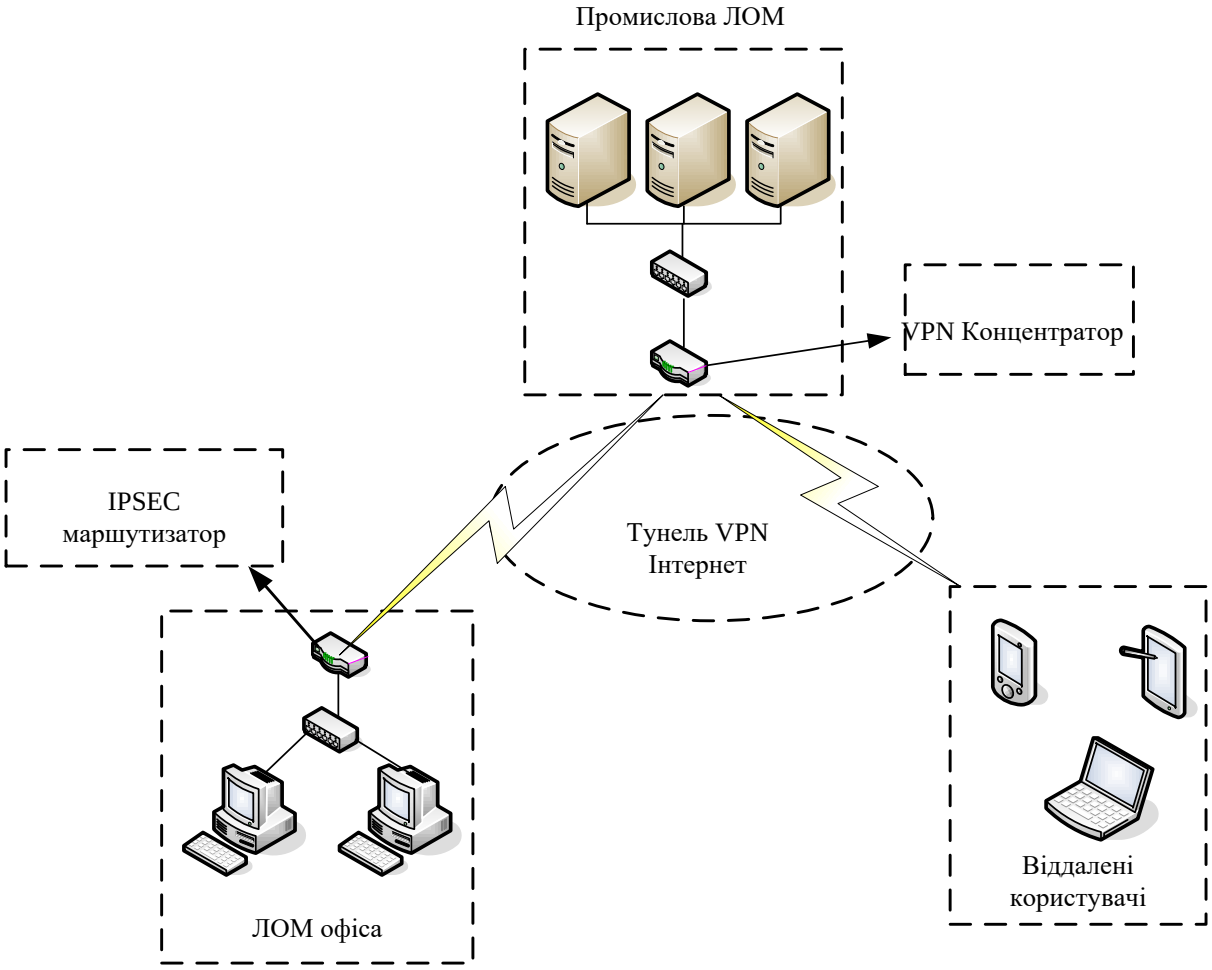


Рисунок 1.5 – Побудова системи на VPN-тунелях IPSEC

Після аутентифікації віддалений користувач потрапляє на сторінку порталу, яка дозволяє отримати доступ до SSL VPN мереж. Всі запити, надіслані браузером включають аутентифікацію Cookies. Сторінка порталу надає всі ресурси, наявні у внутрішній мережі. Наприклад, на сторінці порталу віддалений користувач може завантажити і встановити тонкий клієнт Java аплет (для переадресації TCP-порт) або тунельний клієнт.

Якщо передавальний IPsec-модуль визначає, що пакет пов'язаний з SA, яке передбачає AH-обробку, то він починає обробку. Залежно від режиму (транспортний або режим тунелювання) він по різному вставляє AH-

заголовок в IP-пакет. У транспортному режимі AH-заголовок розташовується після заголовка протоколу IP і перед заголовками протоколів верхнього рівня (Зазвичай, TCP або UDP). У режимі тунелювання весь вихідний IP-пакет обрамляється спочатку заголовком AH, потім заголовком IP-протоколу. Такий заголовок називається зовнішнім, а заголовок вихідного IP-пакета – внутрішнім. При встановленні SA послідовний номер встановлюється в 0, і перед відправкою кожного IPsec-пакета збільшується на одиницю. Крім того, відбувається перевірка –чи не зациклений лічильник. Якщо він досяг свого максимального значення, то він знову встановлюється в 0. Якщо використовується послуга щодо запобігання повторної передачі, то при досягненні лічильника свого максимального значення, що передає IPsec-модуль переустановлює SA. Таким чином забезпечується захист від повторної посилки пакета- приймальний IPsec-модуль буде перевіряти поле SequenceNumber, і ігнорувати пакети щоприходять повторно. Далі відбувається обчислення контрольної суми ICV. Треба зауважити, що тут контрольна сума обчислюється із застосуванням таємного ключа, без якого зловмисник зможе заново обчислити хеш, але не знаючи ключа, не зможе сформувати правильну контрольну суму.

Таке рішення має ряд преваг перед технологіями розглянутими вище, а саме:

− апаратні модулі шифрування;

− професійне стандартизоване виробником обладнання, що використовується для управління трафіком;

− швидкий і безпечний доступ до мережі сервера;

− надійність і безперебійність роботи тунелю;

− високий рівень забезпечення конфіденційності даних за рахунок використання різних методів шифрування;

Із недоліків цієї системи можна виразити тільки один - це висока вартість володіння (придбання, налаштування, обслуговування)

Підсумуємо характеристику розглянутих методів підключення в порівняльній таблиці1.5.

Таблиця 1.5 - Таблиця порівнянь методів підключень

Характеристика	RDP	OpenVPN	IPSEC
Вартість	Відсутня	Невелика.	Висока
Рівень безпеки при передачі даних	Не високий	Задовільний	Високий
Рівень захисту сервера	Не захищений	Захищений	Захищений
Гнучкість рішення	Так	Ні	Так
Надійність рішення	Не висока	Не висока	Висока
Промислове рішення	Ні	Ні	Так

Безліч представлених систем і варіантів показує, що дистанційне керування реалізується в нетривіальних конфігураціях, а концепцію, що цікавить нас,можна оптимізувати різними способами.

Виявлено, що при отриманні віддаленого доступу до інформаційних систем користувач зобов’язаний пройти механізми аутентифікації – процедура перевірки легальності користувача або даних, та авторизації – це надання певній особі або групі осіб прав на виконання певних дій. Провівши аналіз існуючих механізмів аутентифікації користувачів було виділено 3 основних характеристики, якими володіє кожен з них:

− ступінь автоматизації (мається на увазі автоматизація аутентифікації з боку системи, а не користувача) може бути: повною - для аутентифікації гостя непотрібно ніяких додаткових втручань зі сторони адміністратора, або неповною – для аутентифікації гостя необхідне втручання адміністратора;

− пріоритет використання – це те, в якому порядку користувач користується способами аутентифікації;

− використовуваний фактор аутентифікації – аутентифікація являє собою процес порівняння інформації, що надається користувачем, з еталонною.

Метою політики віддаленого доступу є встановлення стандартних норм безпечного віддаленого з'єднання будь-якого хоста з мережею віддалених інформаційних ресурсів. Ці стандартні норми покликані мінімізувати збиток через можливе неавторизоване використання інформаційних ресурсів. До такого збитку відносяться втрата інтелектуальної власності, втрата конфіденційних даних, спотворення іміджу, пошкодження критичних внутрішніх систем.

Ця політика стосується всіх співробітників, постачальників і агентів компанії при використанні ними для віддаленого з'єднання з мережею інформаційних ресурсів, комп'ютерів або робочих станцій, які є власністю компанії або перебувають в особистій власності.

Політика віддаленого доступу:

− намічає і визначає допустимі методи віддаленого з'єднання з внутрішньою мережею;

− істотна у великій організації, де мережі територіально розподілені і простягаються до будинків;

− повинна охоплювати по можливості всі поширені методи віддаленого доступу до внутрішніх ресурсів.

Політика віддаленого доступу повинна визначити:

− які методи вирішуються для віддаленого доступу;

− які обмеження на дані, до яких можна отримати віддалений доступ;

− хто може мати віддалений доступ.

Захищений віддалений доступ повинен бути строго контрольованим. Застосовувана процедура контролю повинна гарантувати, що доступ до належної інформації або сервісів отримають тільки ті люди,що пройшли

перевірку. Управління віддаленим доступом не повинно бути настільки складним, щоб це приводило до виникнення помилок.

Для реалізації основних функціональних компонентів системи безпеки для віддаленого доступу застосовуються різні методи і засоби захисту інформації:

- захищені комунікаційні протоколи;
- засоби криптографії;
- механізми аутентифікації і авторизації;
- засоби контролю доступу до робочих місць мережі і з мереж загального користування;
- засоби боротьби з шкідливими програмами і спамом;
- програми виявлення і запобігання атак;
- засоби централізованого управління контролем доступу користувачів, а також безпечного обміну пакетами даних і повідомленнями будь-яких додатків за відкритими IP-мереж.

1.4 Висновки до розділу

У даному розділі було досліджено які технології використовують для отримання доступу до віддалених джерел інформації. Як саме впливають дані технології на безпеку інформації та як організовують захист при обміні інформацією між клієнтом та віддаленими джерелами інформації. Було досліджено які правила безпеки впроваджуються в технологіях віддаленого доступу. Була розроблена система оцінювання даних технологій та за допомогою неї було оцінено дані технології

Було досліджено методи що використовуються при отриманні віддаленого доступу до інформаційних джерел, виявлено недоліки даних методів, на основі яких прийнято рішення про необхідність розробки нових моделей суб’єктів аутентифікації та авторизації.

2 ДОСЛІДЖЕННЯ ХМАРНИХ ТЕХНОЛОГІЙ ДЛЯ СИСТЕМИ НАВЧАННЯ

2.1 Дослідження поняття динамічних систем

Динамічна система – це об'єкт чи процес, який характеризується своїм станом як сукупністю характеристик у деякі моменти часу, і визначено закон еволюції стану динамічної системи у часі.

Математичне моделювання нелінійних динамічних систем є міждисциплінарним інструментом дослідження різноманітних. При цьому реалізується єдиний методологічний підхід, що дозволяє на основі об'єктивних законів аналізувати рух різноманітних динамічних систем різного рівня складності - від механічних до соціальних.

Основна проблема при математичному динамічному моделюванні системи полягає у розробці моделі, адекватної реальним процесам з прийнятною похибкою.

Для простих механічних систем існують доступні для огляду моделі на основі нелінійних диференціальних рівнянь, що повністю відображають динаміку процесу з урахуванням складні нелінійні ефекти.

Важливим аспектом при побудові моделей динамічних систем є визначення залежностей та коефіцієнтів у рівняннях, що використовуються при побудові моделі. При аналізі найпростіших завдань механіки вид рівнянь повністю визначається постановкою задачі та заданим рівнем точності моделі. Для більш складних динамічних систем визначення коефіцієнтів і залежностей у моделі є нетривіальним завданням.

2.2 Дослідження поняття віртуалізації та її застосувань

Мейнфрейм – головний комп’ютер обчислювального центру з великим об’ємом внутрішньої та зовнішньої пом’яті.

Блейд- сервер – комп’ютерний сервер з компонентами, що об’єднані для збільшення простору.

Системи збереження даних – програмно-апаратні рішення по організації надійного збереження інформаційних ресурсів та надання до них гарантованого доступу.

Мережі зберігання даних – швидкісна комутована мережа передачі даних, що об’єднує сервери, робочі станції, дискові сховища.

Консолідація - об’єднання обчислювальних ресурсів або структур (засобів) керування в одному місці або центрі.

Віртуалізація – абстракція обчислювальних ресурсів, що інкапсулює власну реалізацію.

Віртуальна машина – програмне або апаратне середовище, що приховує існуючий процес або об’єкт від його реального представлення. Повністю ізольований контейнер, що має власну ОС та додатки, подібно фізичному комп’ютеру. Має програмні (віртуальні) ОЗП, жорсткий диск та мережевий адаптер.

Віртуалізація ресурсів фізичного сервера допомагає гнучко розподілити їх між додатками, кожний з додатків бачить лише назначені йому ресурси, та вважає що йому виділений окремий сервер. Таким чином одна потужна машина виконує задачі кількох комп’ютерів. Потрібна щоб абстрагувати програмне забезпечення від апаратної частини.

Віртуалізація ресурсів фізичного сервера служить для їх гнучкому розподілі між додатками, кожен додаток бачить тільки призначені йому ресурси і вважає, що йому було призначено окремий сервер. Таким чином, одна потужна машина виконує завдання декількох комп'ютерів. Призначена щоб абстрагувати програмне забезпечення від обладнання[2].

2.2.1 Переваги технологій віртуалізації

Ефективне використання комп'ютерних ресурсів. Замість 3, 10 серверів, завантажених на 5-20%, можливість використовувати той, який використовує 50-70%. Це економія енергії, а також значне скорочення фінансових вкладень: один високотехнологічний сервер, що виконує функції 5-10 серверів.

Віртуалізація може забезпечити набагато більш ефективне використання ресурсів, оскільки вона об'єднує стандартні ресурси інфраструктури в єдиний фонд і перевищує обмеження застарілої моделі «один додаток на сервер».

Зниження витрат на інфраструктуру: віртуалізація скорочує кількість серверів і пов'язаного з ними IT-обладнання в інформаційному центрі. В результаті зменшується потреба в обслуговуванні, харчуванні і охолодженні матеріальних ресурсів, а IT-відділи витрачають набагато менше грошей.

Скоротіть витрати на програмне забезпечення. Деякі постачальники програмного забезпечення ввели окремі схеми ліцензування спеціально для віртуальних середовищ [2].

Підвищення гнучкості і швидкості реагування системи: віртуалізація пропонує новий спосіб управління IT-інфраструктурою та допомагає IT-адміністраторам витрачати менше часу на повторювані завдання.

При використанні віртуального сервера - можливий запуск відразу на будь-якому обладнанні, а якщо такого сервера немає, можна скачати готову віртуальну машину з встановленим і налаштованим сервером з бібліотек, підтримуваних розробниками гіпервізору (програм віртуалізації).

Несумісні додатки можуть працювати на одному комп'ютері. Коли віртуалізація використовується на одному сервері, ви можете встановити сервери Linux і Windows, шлюзи, бази даних та інші повністю несумісні додатки в єдину невіртуалізірованную систему [2].

Підвищення доступності додатків і безперервності бізнесу: завдяки надійному резервного копіювання та простоїв віртуальних середовищ ви

можете скоротити заплановані простої і забезпечити швидке відновлення системи в критичних ситуаціях.

«Збій» одного віртуального сервера не приводить до втрати інших віртуальних серверів. Крім того, в разі виходу з ладу одного фізичного сервера можлива автоматична заміна на резервний сервер. І це непомітно для користувачів без перевантаження. Це забезпечує безперервність бізнесу.

Можливості легкого архівування. Оскільки жорсткий диск віртуальної машини зазвичай являє собою файл певного формату, розташований на будь-якому фізичному носії, віртуалізація дозволяє легко скопіювати цей файл на носій для резервного копіювання в якості засобу архівування та резервного копіювання всієї віртуальної машини.

Ще одна чудова функція - це можливість повністю завантажити сервер з архіву. І ви можете завантажити сервер з архіву, не руйнуючи поточний сервер, і побачити стан справ за минулий період [2].

Поліпшення управління інфраструктурою: використання централізованого управління віртуальною інфраструктурою скорочує час адміністрування сервера, забезпечує балансування навантаження і «живу» міграцію віртуальних машин [2].

2.2.2 Особливості віртуальних машин

Сумісність. Віртуальні машини сумісні з усіма стандартними комп'ютерами. Як і на фізичному комп'ютері, на віртуальній машині працює власна гостьова операційна система і власні програми. Також він містить всі стандартні для фізичного комп'ютера компоненти (материнська плата, відеокарта, мережевий контролер і т. Д.). Таким чином, віртуальні машини повністю сумісні з усіма стандартними операційними системами, програмами та драйверами.

Віртуальна машина може використовуватися для виконання будь-якого програмного забезпечення, придатного для відповідного фізичного комп'ютера [2].

Ізоляція. Віртуальні машини повністю ізольовані один від одного, як якщо б вони були фізичними комп'ютерами. Віртуальні машини можуть використовувати всі 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 працюють чотири віртуальні машини і одна з них виходить з ладу, це не впливає на доступність трьох інших машин.

Ізоляція - важлива причина набагато більшої доступності та безпеки додатків, що працюють у віртуальному середовищі, в порівнянні з додатками, що працюють в стандартній невіртуалізірованої системі [2]

Інкапсуляція. Віртуальні машини повністю охоплюють обчислювальну середу. Віртуальна машина - це програмний контейнер, який пов'язує або «інкапсулює» повний набір віртуальних апаратних ресурсів, а також ОС і все її застосування в програмний пакет. Інкапсуляція робить віртуальні машини неймовірно мобільними і простими в управлінні.

Незалежність від обладнання. Віртуальні машини повністю незалежні від основного фізичного обладнання, на якому вони працюють. Наприклад, для віртуальної машини з віртуальними компонентами ви можете зробити настройки, які не відповідають фізичним характеристикам базового обладнання.

Віртуальні машини можуть навіть працювати з різними операційними системами (Windows, Linux і т. Д.) На одному фізичному сервері. У поєднанні з функціями інкапсуляції і сумісності апаратна незалежність дозволяє вам вільно переміщувати віртуальні машини з одного комп'ютера на базі k86 на інший без зміни пристроїв, ОС або драйверів [2].

Незалежність від обладнання також дозволяє запускати комбінацію зовсім різних операційних систем і додатків на одному фізичному комп'ютері.

2.3 Дослідження хмарних технологій

Хмарні обчислення необхідні для отримання необхідних обчислювальних можливостей за запитом з мережі. Це динамічно масштабований процес доступу до зовнішніх ресурсів у вигляді сервісу. Такі обчислення дозволяють користувачам занизати складність інформаційних систем при використанні ефективних технологій, що керуються самостійно та мають доступ за першою вимогою[1].

2.3.1 Принцип роботи хмарних обчислень

Замість того, щоб придбати устаткування та власноруч його встановлювати і керувати їм для запуску додатків, можна брати в оренду сервер у будь-якої компанії , що надає подібні послуги, та керувати серверами за допомогою Інтернету. Такі обчислювальні хмари складаються із дуже великої кількості серверів, що розміщені в спеціалізованих дата-центрах.

Зокрема, це уніфікована інфраструктура зберігання даних, яка є невід'ємною частиною структури хмарної архітектури середовища ІКТ, орієнтована на інтегроване сховище даних і управління великими масивами. Головною особливістю, що визначає цю архітектуру, яка забезпечує можливість уніфікації та однорідності її структури, є віртуалізація додатків.

Віртуалізація додатків - це технологія використання і доставки програмного забезпечення (програмних рішень) без його установки на персональний комп'ютер користувача[3].

Обробка і зберігання даних відбувається в дата-центрі, і для користувача робота з хмарними додатками нічим не відрізняється від роботи з програмним забезпеченням, встановленим на його робочому місці.

2.3.2 Апаратне забезпечення хмарних обчислень

Апаратне забезпечення хмарних обчислень може бути різним та постійно еволюціє. Одними з поширених є апаратне забезпечення у вигляді

Blade-серверів – модульна одноплатна система, що має процесор та пам’ять. Дана технологія схожа с системами менфреймів.

Переваги блейд-систем[4]:

- Унікальна фізична конструкція – за допомогою такої конструкцій рішається питання використання таких ресурсів як засоби живлення, охолодження, комутації та керування, знижує складність, що зазвичай характерні для окремих серверів. Така конструкція має шасі що об’єднує сервера, та зв’язок серверів з іншими мережами локальними та глобальними, а також мережами збереження даних.

- Керування та гнучкість систем – керування блейд-системою проходить з централізованого модулю керування та спеціального процесору віддаленого керування на кожному блейд-серверу, та має зручне програмне забезпечення для віддаленого керування системою та живленням.

- Маштабованність – в випадку необхідності дані системи зручно можна масштабувати, додав додаткові шасі. Такі системи є більш зручними завдяки модульній архітектурі при модернізації.

- Висока надійність – в таких системах не потрібно додатково встановлювати обладнання, засоби комутації та мережеві компоненти для резервування для отримання надійності. В системах вже є вбудовані засоби резервування та додаткове живлення. Так само і система охолодження – встановлено кілька вентиляторів, що підтримують систему в робочому стані.

- Заниження витрат – такі системи є більш економічними з боку витрат на енергію та займає мого місця в серверних стійках.

Іншим рішенням є системи та мережі зберігання даних – такі програмно-апаратні рішення по організації надійного збереження інформаційних ресурсів та надання до них гарантованого доступу [4].

Ці системи є не менш надійними, та виділені в окремий вузол. Підключається до серверів за різним типом підключення. Найбільш швидким є підключення за оптичними каналами. Також мають резервування апаратних компонент – живлення, контролерів, адаптерів и тд.

Переваги використання СЗД[4]:

- Надійність та відмово стійкість – повне або часткове резервування усіх компонент системи, а також використання систем моніторингу та систем сповіщення;
- Доступність даних – використання функцій зберігання цілісності даних та додавання апаратури і ПЗ в неперервно робочу систему;
- Засоби керування та контролю – використання повного моніторингу на рівні апаратури за технологією діагностики продуктивності та керування системою через web-інтерфейс або командний рядок.
- Продуктивність – забезпечується кількістю жорстких дисків, об’ємом кеш-пам’яті, обчислювальною міцністю процесорної системи, кількістю внутрішніх та зовнішніх інтерфейсів.
- Маштабованість – є можливість нарощування кількості жорстких дисків, об’єма кеш-пам’яті. Такі функції забезпечують гнучкість в проектування мережі зберігання даних.

Мережі зберігання даних – це швидкісна комутована мережа передачі даних, що об’єдную сервери, робочі станції, дискові сховища. Обмін даними відбувається по протоколу оптичного зв’язку та гарантує передачу даних на великі відстані.

Переваги мереж зберігання даних[4]:

- забезпечують високу продуктивність для збереження та передачі даних;
- забезпечують масштабованість та розширення підсистем зберігання, що дає змогу легко використовувати більш рані версії обладнання з новими пристроями зберігання.
- гнучкість – дозволяє сумісно використовувати системи зберігання даних та спростити адміністрування та додає гнучкості, оскільки кабелі та дискові масиви не потрібно фізично транспортувати та перекомутувати від одного сервера до іншого.

- централізоване завантаження – дає можливість завантажувати сервера з мережі зберігання.

- відмово стійкість – дозволяють швидко та ефективно відновити працездатність після збоїв за допомогою віддаленої ділянки з другорядним пристроєм зберігання. В такому випадку використовується реплікація що реалізується на рівні контролерів масивів.

- керування підсистемою зберігання даних централізоване.

Використання консолідації ресурсів дозволить покращити керованість системами за рахунок більш актуальною та повної інформації о функціонуванні. Консолідація може бути:

- серверів – переміщення додатків що розташовані на різних серверах в один кластер;

- системи зберігання – сумісне використання централізованої системи зберігання даних кількома вузлами;

- додатків – розміщення кількох додатків на одній робочій станції.

Можна використовувати два типи консолідації – фізичну та логічну, в залежності від потреб. Фізична, якщо розміщення серверів на єдиної платформі, та логічна - централізоване керування.

Фізична консолідація дозволяє підняти рівень фізичного захисту серверів. В ЦОД використовується більш продуктивне обладнання, що не ефективно використовувати у власному розпорядженні. Централізація спрощує використання стандартизованих конфігурацій та процесів керування, створення рентабельних систем резервного копіювання для відновлення даних після збоїв.

Логічна консолідація – пере налаштування системи керування IT-інфраструктури. Необхідна для збільшення маштабованності та керування складними розподіленими обчислювальними системами та об’єднання сегментів корпоративної мережі.

Переваги: вивільнення апаратних ресурсів, які можливо використовувати і на інших сегментах інформаційної системи, спрощена та логічна структура керування IT-інфраструктурою.

Гемогена консолідація – перенесення одного великого додатка, що раніше виконувався на кількох серверах, на один сервер більш потужний.

Гетерогенна консолідація – об’єднує різні додатки, що раніше завантажувались на різних комп’ютерах, на один комп’ютер. Збільшує масштабованість сервісів та повноцінно задіє в роботи системні ресурси.

2.4 Характеристики хмарних обчислень

Самообслуговування за запитом. Користувач вибирає, які обчислювальні потужності і ресурси використовувати (наприклад, мережеве сховище, обсяг оперативної пам'яті, бази даних, час обробки), при необхідності автоматично змінює цей набір без згоди виробника [3].

Висока еластичність (гнучкість) сервісів. Обчислювальну потужність можна легко зменшити або збільшити в залежності від потреб користувача. При великому навантаженні на сервіс кількість ресурсів швидко збільшується, при зниженні навантаження - ресурси вивільняються[3].

Можливість об'єднувати ресурси. Обчислювальні ресурси хмарних провайдерів згруповані в пули з можливістю динамічного розподілу фізичних та віртуальних ресурсів між кінцевими користувачами. Використання сучасних технологій віртуалізації дозволяє постачальнику хмарних послуг легко збільшувати потужності і замінювати обладнання без шкоди для продуктивності і надійності [3].

Веде облік витрат ресурсів і оплати при використанні. Користувач сплачує тільки за фактично спожиті послуги (наприклад, за обсяг переданої інформації, пропускну здатність і т. д.) [3].

Технологічність. Постачальники хмарних даних використовують більш сучасні інноваційні технології в центрах обробки даних. Ці технології

дозволяють автоматично оптимізувати використання обчислювальних ресурсів і знизити витрати на обслуговування обладнання в порівнянні з аналогічними витратами в навчальних закладах [3].

Стійкість до помилок і висока доступність. Центри обробки даних хмарних обчислень забезпечують надійну розподілену мережу, вузли якої можуть бути розташовані в різних частинах світу. Відмовостійкість вище, ніж в локальній мережі, оскільки забезпечується багаторазовим резервуванням і кваліфікованим обслуговуванням технічного персоналу [3].

2.5 Класифікація хмарних обчислень

Модель обслуговування визначає рівень автоматизації процесу IT-інфраструктури [4]. Існують наступні моделі хмарних сервісів:

Інфраструктура як послуга (IaaS, Infrastructure as a Service). Користувач може самостійно проектувати і керувати своєю IT-інфраструктурою в хмарі: створювати віртуальні мережі, додавати віртуальне обладнання (сервери, сховище, бази даних), встановлювати необхідне прикладне програмне забезпечення та операційні системи, або використовувати хмару, як якщо б це була справжня IT-інфраструктура навчального закладу. Найвідоміші рішення IaaS: Amazon CloudFormation, Google Compute Engine, Windows Azure [5].

Платформа як послуга (PaaS, Platform as a Service). На цьому рівні постачальник хмарних послуг надає користувачеві доступ до операційних систем, систем управління базами даних, інструментів розробки і тестування. Таким чином, користувач хмарних сервісів отримує можливість і ресурси самостійно створювати, тестувати і управляти програмним забезпеченням.

Вся інформаційна інфраструктура (комп'ютерні мережі, сервери та системи зберігання) управляється постачальником. Найбільш відомі PaaS-сервіси: Google App Engine (для розробки програмного забезпечення на Java,

Python); Windows Azure (для ASP.NET, PHP); Cloud Foundry (мови програмування Java, Ruby, Scala) [5].

Програмне забезпечення як послуга (SaaS, software as a service). На цьому рівні постачальник надає користувачам хмари найсучасніше програмне забезпечення. Всі дані зберігаються в хмарі, і користувачеві потрібен тільки веб-браузер для доступу до них. Це найбільш цікавий вид хмарних обчислень для освітніх установ, оскільки він не вимагає додаткових витрат на установку і настройку програмного забезпечення, як це необхідно при використанні IaaS і PaaS.

Також слід мати на увазі, що в більшості випадків плата за використання програмного забезпечення в рамках SaaS розраховується на основі кількості користувачів і не передбачає так званих корпоративних ліцензій, які дозволяють вам використовувати послугу для будь-якої кількості користувачів без обмежень. Рішеннями SaaS для освітніх установ є Google Apps для освіти і Microsoft Office 365 для освіти[5].

Вони включають в себе функції офісного пакету (робота з документами, електронними таблицями і презентаціями), засоби зв'язку (електронна пошта, календарі, обмін миттєвими повідомленнями) і інструменти для ефективного представлення інформації (у формі статичних презентацій, відео або інтерактивних додатків).

2.6 Висновки до розділу

В другому розділі проведено дослідження та виявлення понять динамічних систем, понять віртуалізації та її типів, а також, хмарні сервіси. Визначена основна характеристика хмарних обчислень та наведена їх класифікація. Дослідження проведено з ціллю визначити можливість використання технологій в учбовому процесі.

3 МОДЕЛІ ПРИВАТНОЇ УЧБОВОЇ ХМАРИ ВІДДАЛЕНОЇ ЛАБОРАТОРІЇ

Виходячи з аналізу існуючих технологій, які здатні сприяти технологічному розвитку віртуальних учбових лабораторій, можна зробити висновок, що для наступних напрямків лабораторних робіт, доцільно використовувати приватне хмарне середовище:

- IT напрямки (мережі, програмування, автоматизовані процеси тощо);
- хімія (емуляція і моделювання процесів);
- фізика (емуляція і моделювання процесів);
- біологія (моделювання);
- статистика (аналіз великих даних);
- художні напрями (дизайн, малювання тощо);
- музичні напрямки (обробка звуків тощо);
- інше.

Особливостями більшості лабораторних стендів для таких робіт в реальних лабораторіях є наступні характеристики:

- недостатня кількість стендів для всіх учасників учбового процесу;
- різноманітність лабораторних стендів;
- необхідність в наявності певних елементів учбового стенду;
- відсутність постійної необхідності в наявності стенду.

Тому, в даній роботі проведено дослідження моделей, використання яких дозволяє динамічно створювати об’єкти лабораторних стендів виключно за потребою та створювати гнучкі структури взаємодії між створеними об’єктами.

Для цього необхідно розробити ряд моделей, котрі описують процеси створення, зберігання та взаємодії об’єктів учбових лабораторій.

Моделі, які описують динамічне віртуальне середовище, повинні сприяти виконанню таких вимог до учбових проектів, як:

– простота для кінцевих користувачів. Для викладачів повинні бути прості механізми створення учбових проектів, контролю їх виконання, модернізації наявних проектів. Студенти мають можливість простого та зручного доступу до учбових матеріалів та активності (лабораторних робіт, практичних завдань, розрахункових та дослідницьких робіт тощо);

– цілісність та доступність даних. Приватна учбова хмара повинна забезпечити незмінність учбових матеріалів, які завантажуються викладачем та отримують студенти в результаті виконання активності. Дані повинні бути доступні для викладачів та студентів за запитом;

– конфіденційність даних. Моделі учбових даних повинні забезпечити ізоляцію цих даних між користувачами. Без дозволу викладачів їх учбові матеріали не повинні бути доступні іншим, метадані проектів повинні бути доступні тільки розробнику проекту. Хід виконання активності студентом доступний тільки йому, результати виконання доступні на основі побажань студентом.

3.1 Модель мережевої структури лабораторної роботи

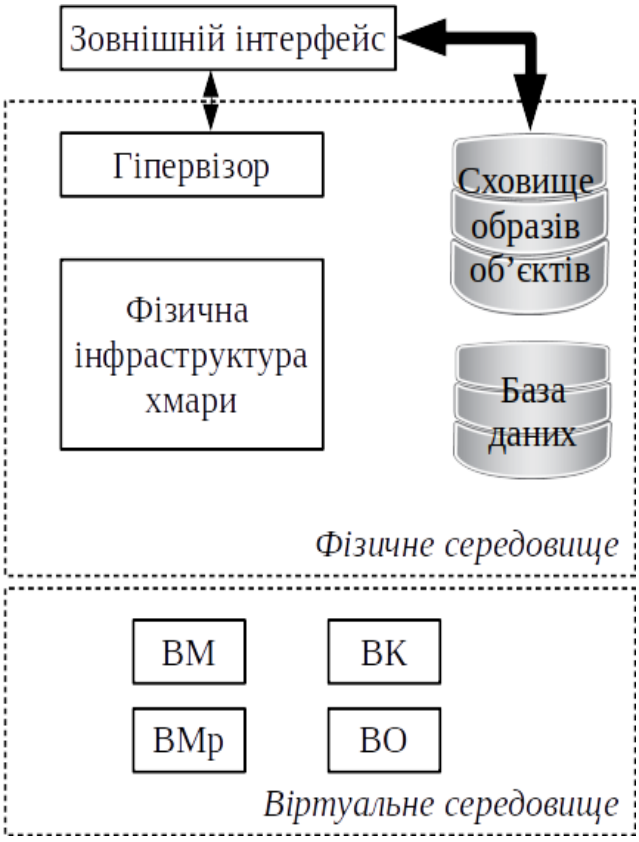
Модель мережевої структури передбачає опис динамічних взаємозв’язків між тимчасовими об’єктами віртуальних лабораторних стендів на час проведення роботи. Також, об’єктами віртуальної лабораторії можуть бути не тільки стенди, але й інші активності, які передбачені учбовим процесом.

Для реалізації цих цілей задовольняє використання приватної мережевої хмари, де кожна лабораторна робота або інша учбова активність представлена у вигляді окремого проекту.

Модель мережевої структури приватної хмари налічує наступні елементи:

– множина хостів мережі;
– комунікаційне обладнання;
– множина віртуальних об’єктів (стендів);
– віртуальні комутатори;
– сховище образів об’єктів;
– база даних системи керування;
– гіпервізор керування проектів.

Структурна модель проекту представлена на рисунку 3.1



ВМ - Віртуальна Машина, ВК - Віртуальний Комутатор

ВМр - Віртуальний маршрутизатор, ВО - Віртуальні Об’єкти

Рисунок 3.1 - Модель мережевої структури віртуальної лабораторії

Фізичні хости, комутатори та маршрутизатори формують фізичну

інфраструктуру приватної учбової хмари. На основі даної інфраструктури формуються віртуальні середовища проектів (лабораторних) для виконання студентами учбових активності.

Віртуальними об’єктами можуть виступати наступні сутності:

– віртуальні машини (операційні системи, програмування, комп’ютерні мережі тощо);

– віртуальні контролери (автоматизовані системи, електромеханіка, мережі тощо);

– віртуальне комунікаційне обладнання (комутатори, маршрутизатори, гіпервізори, шлюзи тощо);

– віртуальні дослідні стенди (біологія, хімія, фізика, електротехніка, електромеханіка тощо);

– інші учбові об’єкти, які піддаються віртуалізації.

Віртуальні об’єкти створюються за запитом виключно на певний час активності та знищуються після її завершення. Це дозволяє використовувати одну й ту ж інфраструктуру для різних типів завдань різних дисциплін та різних напрямів освітніх програм.

Віртуальні комутатори створюються для створення зв’язків між віртуальними об’єктами проекту, якщо простого з’єднання з єдиним центром для логіки активності недостатньо. На основі віртуальних комутаторів можна створювати складні ієрархічні віртуальні структури взаємозв’язків між об’єктами проектів, фільтрувати доступ між об’єктами та інші задачі.

Сховище образів об’єктів використовується для зберігання підготовлених сутностей для виконання певних учбових активності студентами. Сховище повинно забезпечити зберігання потрібної кількості образів, забезпечувати цілісність та доступність образів.

Гіпервізор керування виконує наступні функції:

– керування фізичною інфраструктурою;

– створення проектів на основі образів;

- контроль віртуальної структури проекту;
- знищення проекту.

Керування фізичною інфраструктурою передбачає підготовку мережевих об’єктів для конкретного проекту:

- розгортання образів об’єктів на фізичних хостах. Створюються віртуальні машини, які реалізують поставлену учбову задачу або задачі;
- створення віртуальних мереж. Організація відокремленої мережі для об’єднання створених віртуальних об’єктів в єдиному середовищі;
- контроль доступності та цілісності створених віртуальних об’єктів.

Створення проектів виконується за рахунок вибору з бази даних опису проекту конкретної активності, де описуються типи віртуальних об’єктів, їх кількість та спосіб їх об’єднання.

Після визначення структури проекту, вибираються потрібні образи зі сховища та розгортаються на фізичних хостах. Фізичні комутатори налаштовуються на створення ізольованої мережевої структури для цих об’єктів.

Контроль віртуального середовища передбачає контроль доступності кожного об’єкту, зв'язності мережі, цілісності сервісів віртуальних об’єктів.

Видалення проекту виконується після завершення активності за командою студента, визначеного проектом таймеру або викладачем при, наприклад, достроковому виконанні активності.

3.2 Модель керування шаблонами структур

Враховуючи, що виконання лабораторних робіт, в умовах створення динамічної структури, передбачає створення тимчасових об’єктів, які потрібні в ході виконання лабораторних робіт, необхідно організувати

наступні функції, для роботи з ними:

- створення шаблонів стендів;
- створення шаблонів окремих складових стендів;
- зберігання шаблонів;
- ініціація шаблонів;
- модифікація шаблонів;
- видалення шаблонів.

З перелічених вище пунктів можна зробити висновок, що основою даного проекту є дві складові - мережева структура та шаблони її об’єктів. Для зазначених функції та елементів необхідно створити відповідні моделі, які можна використовувати для опису процесів роботи системи віддаленої лабораторії.

Кожний шаблон описує учбовий проект, котрий може бути окремим об’єктом або множиною об’єктів та опису їх взаємодії. Виходячи з цього можна сформувати, відповідно, дві моделі опису шаблонів проекту.

3.2.1 Модель окремого об’єкту

Дана модель описує окрему одиницю проекту, або сама може виступати в якості проекту. Наприклад, віртуальна модель лабораторного стенду може використовуватись в якості дослідницького проекту, або бути частиною комплексного стенду.

Модель повинна забезпечити опис множини різноманітних образів віртуальних об’єктів, які зберігаються в хмарному сховищі

$$I = \langle OS, Res, App, Net, Srv \rangle$$

де, OS - множина операційних систем яка може бути встановлена на віртуальному об’єкті;

Res = <CPU, MEM, DISK> - множина ресурсів віртуального об’єкту, таких як процессор, ОЗП, дисковий простір;

Srv - множина додаткових сервісів, які не входять в перелік

стандартних для обраної операційної системи;

Net <IP, MAC, BW> - множина параметрів мережної взаємодії об’єкту;

App <A1, A2, …, An> - множина додаткового програмного забезпечення специфічного для проекту.

З обов’язкових елементів моделі можна зазначити наступні:

- операційна система є обов’язковим системним елементом кожного віртуального об’єкту, так як є основним прикладним програмним забезпеченням для моделювання сутностей, процесів тощо;
- для реалізації програмних симуляцій необхідні відповідні до задачі системні ресурси, які виділяються при розгортанні образу об’єкту. Виділені ресурси є фіксованими, тому необхідно заздалегідь визначити необхідні значення для кожного проекту.

Ці дві множини описують мінімальний образ об’єкту учбового процесу в мережі, який використовується студентами для виконання активностей.

Всі інші параметри є додатковими та розширюють можливості об’єкту в рамках учбового процесу. Наприклад, можна зберігати образ, котрий реалізує симуляцію хімічного процесу, що вимагає, окрім мінімального образу, мати наявність спеціалізованого програмного забезпечення та відповідних сервісів для цього програмного забезпечення.

Також, якщо логіка роботи передбачає розподілену систему об’єктів хмари або зв’язок із зовнішніми ресурсами, необхідна наявність мережевих з’єднань для об’єкту що розгортається.

3.2.2 Модель комплексного проекту

Модель комплексного проекту описує складаний проект, частинами якого є моделі окремих проектів та система зв’язків між ними.

Модель описує кількість образів в проекті, кількість віртуальних комутаторів та взаємозв’язок між ними

$$M = <I, S, C>$$

I = {I1, I2, …, In} - множина образів комплексного проекту,

S = {∅, S1, S2, …, Sm} - множина віртуальних комутаторів,

C = {∅, <Ii, Sk>, <Ii, Ij>} - множина пар зв’язків між об’єктами проекту

Очевидним фактом є те, що, якщо множини віртуальних комутаторів та зв’язків є порожніми множинами, дана модель описує окремий об’єкт з попереднього пункту.

Класичним прикладом комплексного проекту є множина образів (більше одного), один віртуальний комутатор, та зв’язки кожного віртуального об’єкту з віртуальним комутатором. Проте існують проекти, де взаємодію між об’єктами описують пари віртуальних образів без використання комутаторів (хімічні симуляції, біологічні симуляції тощо).

3.2.3 Підготовка та зберігання образів

З розібраного вище можна зробити висновок, що кожна учбова активність базується на процесі розгортання образів об’єктів, які реалізують певні учбові завдання. З цього можна зробити висновок, що для підготовки образів необхідні наступні дії (рисунок 3.2):

– підготовка образу. Викладач формує моделі, симуляції, віртуальні об’єкти, які необхідні для реалізації завдання для студентів;

– ініціалізація образів. Після підготовки образів викладач реєструє їх в системі (гіпервізорі віртуального середовища) для реалізації їх доступності в проектах;

– зберігання образів. Створені образи зберігаються в виділеному сховищі образів;

– підготовка проекту. Викладач створює проект, описуючи наявні в ньому образи об’єктів та зв’язків між ними;

– ініціалізація проекту. Викладач реєструє створений проект в гіпервізорі віртуального середовища хмари;

– зберігання проекту. Створений опис проекту зберігається у базі

даних.

– публікація проекту. Викладач публікує створений проект на учбовій платформі, де він стає доступним для виконання студентами.

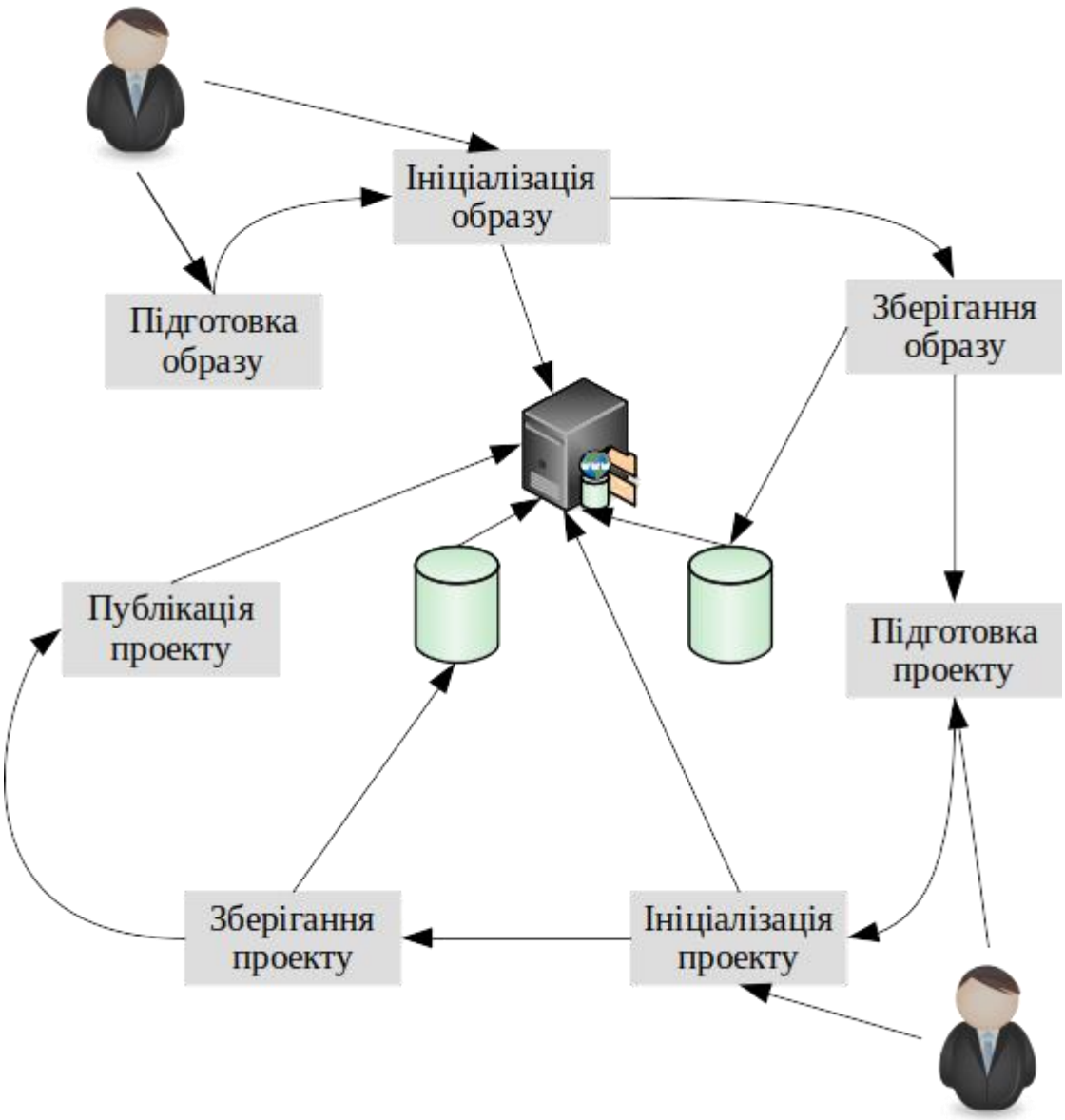


Рисунок 3.2 - Інформаційна модель створення учбового проекту

Зберігання реалізується на виділеному сховищі приватної хмари. Система зберігання забезпечує доступність та цілісність завантажених образів за рахунок реплікації інформації між серверами зберігання або дисками на сервері.

3.3 Модель контролю віртуальних проектів

При створенні динамічних структур для виконання лабораторних робіт, необхідно забезпечення виконання наступних функції контролю:

- контроль доступності вузлів;
- контроль зв’язності вузлів;
- навантаження вузлів;
- стан вузлів.

Контроль доступності вузлів передбачає перевірку активності вузлів. Система гіпервізору перевіряє, чи всі віртуальні об’єкти проекту включені та готові до роботи. В іншому випадку, система повинна сповістити викладача або адміністратора про проблему.

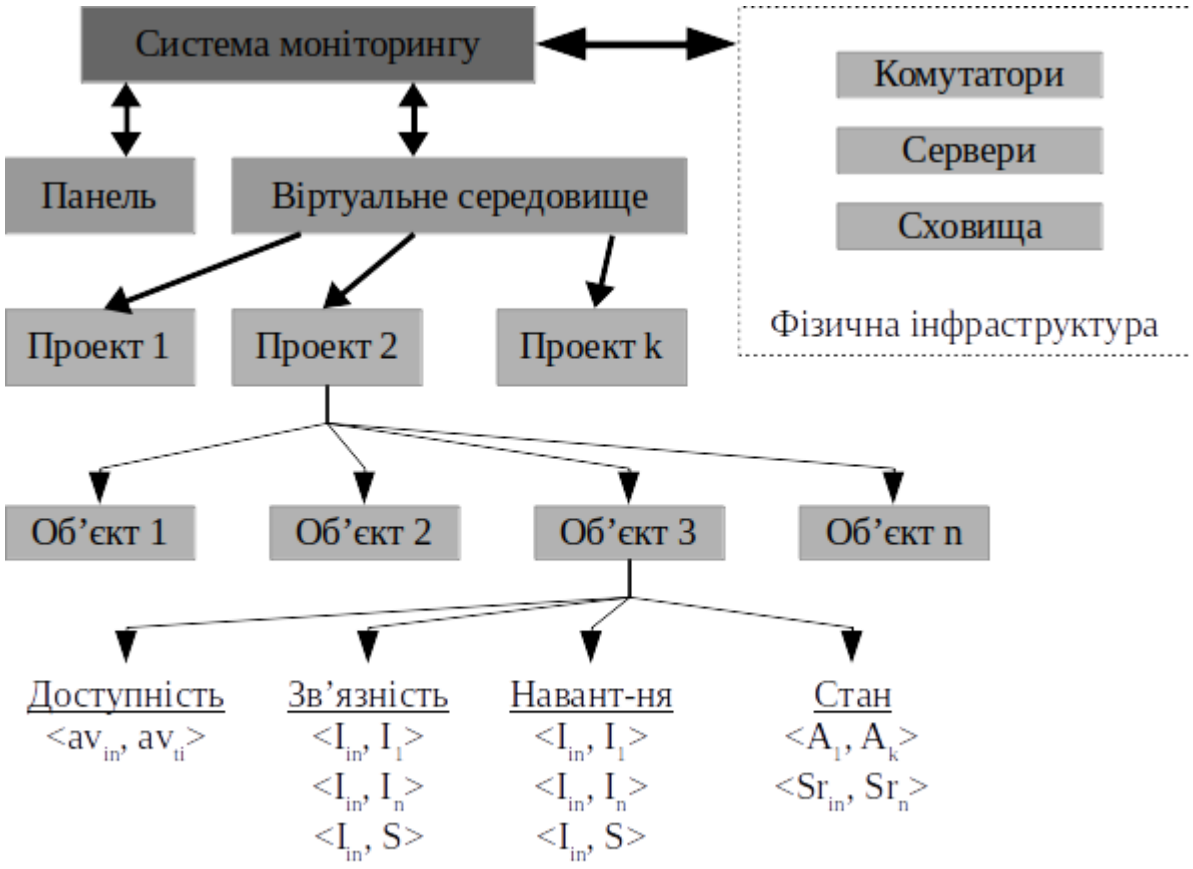


Рисунок 3.3 - Структурна модель контролю віртуального проекту

Контроль зв’язності передбачає перевірку активних віртуальних об’єктів на предмет можливості здійснювати передачу даних один одному та з зовнішніми системи.

Навантаження вузлів передбачає контроль віртуальних об’єктів з боку використання системних ресурсів, таких як навантаження на процесор та дискову систему, використання оперативної пам’яті.

Під контролем стану вузлів розуміється контроль сервісів на активному вузлі, зокрема:

– контроль активності мережевих сервісів віртуального об’єкту;

– контроль сервісів, які додатково встановлені викладачем на етапі формування образу;

– контроль застосувань, які розгорнені на образі віртуальної машини.

На основі даної моделі контролю проектів, можна сформулювати загальну модель контролю учбових проектів

$$MI = \{<ID_{pi}, ID_{oj}>\}$$
$$ID_{oj} = \{<av_j, A_j> \}$$
$$ID_{pi} = \{ ID_{oi}, S_k, Sr_n \}$$

Таким чином, моделі контролю дозволяють здійснювати спостереження за наступними сутностями учбового процесу:

– моніторинг створеної моделі лабораторного стенду;

– контроль виконання лабораторної роботи студентом;

– контроль виконання лабораторної роботи викладачем.

Дані дії дозволяють отримувати інформацію про перебіг активності та своєчасної реакції на непередбачені події в ході виконання завдань.

3.4 Модель даних лабораторних робіт

Виконання студентами лабораторних робіт пов’язано з роботою з певними даними:

- викладачі можуть використовувати певні дані в якості початкових даних для завдань або експериментів;
- студенти, в результаті виконання лабораторних завдань, отримують проміжні або кінцеві дані.

Для забезпечення учбового процесу, необхідно забезпечити зберігання цих даних для їх подальшої обробки. Для цього необхідно забезпечити:

- виділене сховище даних;
- індексацію даних;
- множину метаданих;

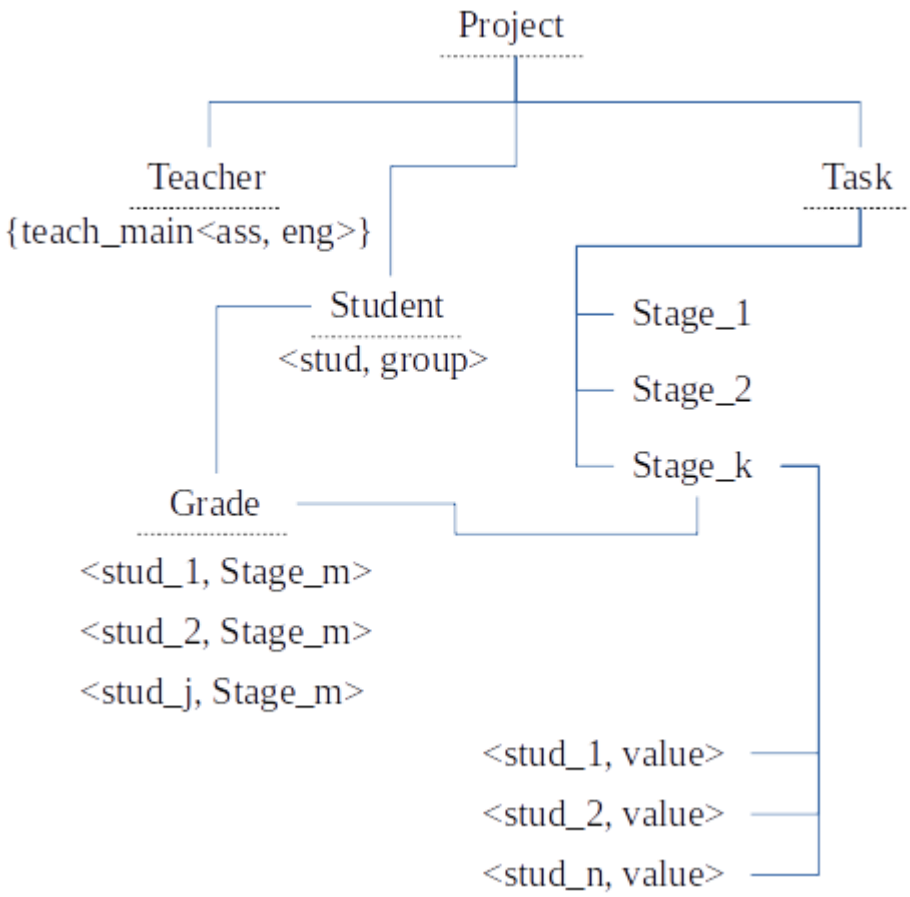


Рисунок 3.4 - Ієрархічна модель учбових даних

Для реалізації розподіленого та авторизованого доступу до даних, які стосуються проектів, в досліджені пропонується модель, яка описує ієрархію інформаційних ресурсів кожного проекту.

Кожному проекту присвоюється унікальний ідентифікатор, котрий асоціюється з викладачем, котрий створив даний проект. Зараховані до проекту студенти, також отримують асоціацію з ним.

Кожний проект складається складається із завдань, які студент повинен виконати. Завдання розділяються на етапи, кожний з яких оцінюється окремо. Моделлю допускається, що проект може складатися з одного завдання, а завдання, відповідно, може складатися з одного етапу.

Кожний етап асоційований зі студентом, який його виконує та значень, які студенти отримують в результаті виконання, що в свою чергу, дозволяє створювати елемент оцінок, які асоціюються з відповідним студентом та етапами, які він виконав.

Для запобігання фальсифікацій з боку студентів, необхідно забезпечити неможливість впливу на результати виконаних активності, що зумовлює наявність функції імпорту даних з проекту до бази даних. Даний імпорт може виконуватись автоматично, або студент може ініціювати цю дію самостійно.

Таким чином, модель описує наступні сутності:

- викладачі, які створюють проект або використовують його. Якщо проект повинен бути доступним для інших викладачів, їм надається спеціальний маркер асоціації;
- студенти. Отримують доступ до проекту за маркером окремого доступу або груповим маркером;
- завдання, які складають логіку виконання проекту;
- етапи, які описують активності кожного завдання;
- оцінки етапів та загальна оцінка виконання проекту.

Кожна сутність зберігається в базі даних учбової хмари та доступна

виключно власникам проекту, що унеможливлює втручання в результати виконання проектів.

3.5 Рекомендації щодо впровадження моделей

На основі запропонованих моделей можна зробити рекомендації щодо їх впровадження для наступних сервісів учбової приватної хмари:

- сховище проектів та сховище для бази даних проектів;
- формування шаблонів проектів та об’єктів;
- моніторинг учбових проектів;

Сховище проектів необхідно для зберігання даних проектів, які можна представити як об’єкт. Серед таких типів даних можна зазначити наступні:

- документи;
- зображення;
- звукові файли;
- відеофайли;
- інше.

Об’єктне сховище дозволяє зберігати дані в плоскому адресному просторі, що спрощує доступ до них для гіпервізора системи та зменшує загальне навантаження на систему, за рахунок індексації об’єктів (рисунок 3.5)

В якості системи зберігання пропонується використовувати протокол CEPH, який забезпечує зберігання даних з наступними характеристиками:

- зберігання в плоскому адресному просторі;
- індексацію об’єктів зберігання;
- реплікацію даних за замовчуванням;
- можливість самостійно відновлюватись.

Система, яка забезпечує життєвий цикл шаблонів, повинна мати простий інтерфейс та зрозумілу мову опису для як для людини, так і для

гіпервізора, який обробляє інформацію для створення проекту.

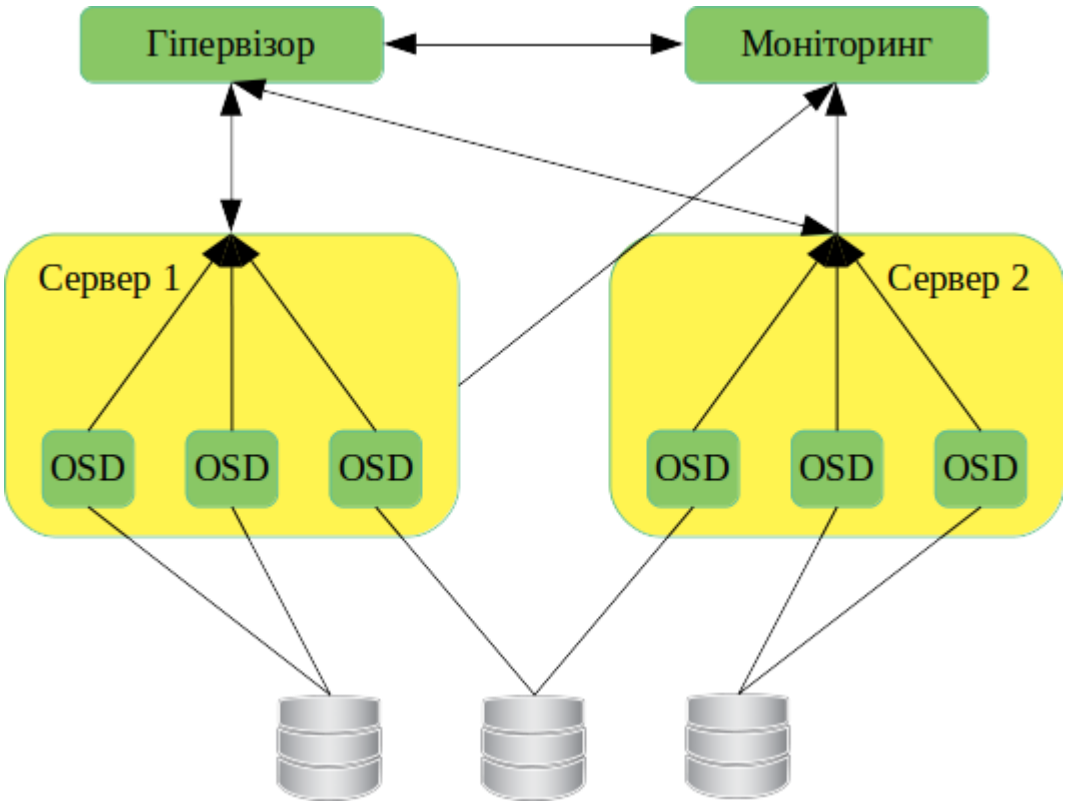


Рисунок 3.5 - Реалізація моделі зберігання даних

Для цих цілей можна використовувати два формати опису: XML розмітку або YAML файли. Кожний з цих форматів дозволяє описувати об’єкти у структурованому вигляді із вказанням усіх необхідних параметрів розгортання.

3.6 Висновки до розділу

В ході дослідження моделей було запропоновано множину елементів, які дозволяють описувати динамічну систему учбової хмари, робота якої спрямована на економію обчислювальних ресурсів із зберіганням доступу студентів до учбового контенту в гнучкому стилі розкладу занять.

Серед основних переваг запропонованих моделей визначені наступні:

– модель мережевої структури дозволяє створювати учбові

практичні та дослідницькі проекти в динамічному середовищі;

– зберігати результати проведених активності для подальшого аналізу та оцінювання;

– створювати та зберігати шаблони учбових об’єктів;

– здійснювати поетапний контроль виконання завдань студентами;

– здійснювати комунікацію між учасниками проекту.

Розроблені моделі вимагають реалізації у вигляді приватної хмари та віртуальної лабораторії. Практична реалізація запропонованих моделей дозволяє:

– скоротити витрати на впровадження лабораторного обладнання для учбових занять;

– розширити спектр активності, які викладачі можуть впроваджувати в учбовий процес;

– розширити спектр дослідницьких задач для студентів та викладачів.

Впровадження даних моделей та технологій дозволяє отримати наступні результати:

– цикл лабораторних робіт від розробки до впровадження скоротився на 9%;

– час перевірки та оцінювання викладачем активності скоротився на 13%;

– кількість можливих активності розширилася на 21%.

ВИСНОВКИ

В ході проведення дослідження було розроблено моделі інформаційних сервісів, які дозволяють спростити процес організації учбового процесу, за рахунок наступних факторів:

– використання шаблонів для учбових робіт дозволяє розширювати перелік активності, які викладачі планують проводити зі студентами;

– використання моделей контролю дозволяє автоматизувати процес проведення практичних робіт студентами;

– використання моделей, заснованих на хмарних системах віртуалізації дозволяє створювати динамічні учбові системи, які використовуються за потребою, і не вимагають постійного існування відповідної інфраструктури;

– також, використання динамічних систем дозволило скоротити апаратні витрати на організацію учбових систем, таких як лабораторні стенди, комп'ютерні мережі, дослідницькі об'єкти тощо.

На основі розроблених моделей реалізовано низку технічних рішень, які організують життєвий цикл практичної роботи для студентів, а саме:

– створення шаблону активності для студентів;

– виділення інформаційних та апаратних ресурсів для виконання роботи;

– контроль апаратно-програмного забезпечення в ході виконання активності студентами;

– контроль даних дозволяє виконувати викладачем оцінювання та відслідковування активності студентами в режимі он-лайн.

Серед переваг системи можна визначити наступні фактори, які вплинули на організацію учбового процесу:

– використання технологій хмарної віртуалізації дозволило забезпечити скорочення апаратних витрат на реалізацію занять із збільшенням кількості студентів;

– використання шаблонів дозволило розширити перелік завдань для студентів, які викладач може проектувати;

– використання динамічних учбових систем дозоляє здійснювати учбову діяльність на основі гнучкого графіку, замість строго детермінованого.

Серед недоліків системи можна зазначити наступні, які залежать як від об’єктивних факторів так і суб’єктивних:

– кількість та технічна складність учбових активності залежить від апаратних можливостей лабораторного обладнання;

– підтримка програмно-апаратної інфраструктури системи потребує професійної підтримки, яку не завжди здатні забезпечити штатні інженери учбових закладів;

– доступність приватної учбової хмари залежить від комунальних особливостей учбового закладу.

Серед наступних етапів розвитку представлено дослідження можна зазначити необхідність забезпеченні заходів, направлених на доступність та безпеку учбової хмари, спрощення адміністрування системи за рахунок автоматизації системних процесів.

Технічне рішення, яке реалізує результати даного дослідження рекомендується впроваджувати в будь-якому учбовому закладі, де передбачено велику кількість студентів які залучені до виконання лабораторних, практичних або дослідницьких робіт.

ПЕРЕЛІК ПОСИЛАНЬ

1. 2 М. В. Мар’Єнко Наукові платформи та хмарні сервіси, їх місце у системі наукової освіти вчителя – 2019 - [Електронний ресурс]. URL: https://cyberleninka.ru/article/n/naukovi-platformi-ta-hmarni-servisi-yih-mistse-u-sistemi-naukovoyi-osviti-vchitelya (дата звернення: 15.03.2021).

2. 6 Л.Ф. Зиангирова Технологии облачных вычислений – 2015 - [Електронний ресурс]. URL: https://intuit.ru/studies/courses/673/529/lecture/11917?page=3 (дата звернення: 05.04.2021).

3. 8 Илья Клементьев, Владимир Устинов. Введение в облачные вычисления. Лекция – 2020 - [Електронний ресурс]. URL: http://dls.kherson.ua/dls/library/LibdocView.aspx?id=da2090b8-ed73-45d0-a5c2-55e307d63bd8 (дата звернення: 17.04.2021).

4. 9 Хмарні обчислення. [Електронний ресурс]. URL: http://integritysys.com.ua/solutions/pricatecloud-solution/ (дата звернення: 20.04.2021).

5. 10 Л.Ф. Зиангирова Технологии облачных вычислений. Лекция 1: Введение. Структура веб-технологий – 2015 - [Електронний ресурс]. URL: https://intuit.ru/studies/courses/606/462/lecture/10378?page=3 (дата звернення: 25.04.2021).

6 IBM Forms documentation, 09.07.2009. – 245 с. [Електронне джерело] Режим доступу: https://www.ibm.com/support/knowledgecenter/en/ SSS28S_8.2.0/ welcome/IBM-Forms-welcome.html

7 Парк Дж., Маккей С. Збір даних в системах контролю і управління. Практичний посібник, ТОВ "Група ІДТ", 2006. – 309 с.

8 А.А. Барсегян, М.С. Купріянов, В.В. Степаненко, І.І. Холод Методи і моделі аналізу даних: OLAP і DataMining, Пітер, 2015. – 1090 с.

9 M. Just. Designing authentication systems with challenge questions. In L. F. Cranor and S. Garfinkel, editors, Security and Usability: Designing Secure Systems that People Can Use, сторінки 143–155, Sebastopol, CA, 2005. O'Reilly Media, Inc. [Електронне джерело] Режим доступу: https://pdfs.semanticscholar.org/fbfb/c601e582f904decf2f739a4e1d41ee86ec0d.pd