



## АНОТАЦІЯ

Дипломна робота складається з пояснювальної записки та графічної частини (ілюстративний матеріал –слайди). Об'єм графічної частини дипломної роботи становить 10 слайдів. Об'єм пояснювальної записки складає 105 друкованих сторінок формату А4 (210×297), об'єм додатків –14 друкованих сторінок формату А4. Дипломна робота складається з восьми розділів, в яких нараховується 47 рисунків та 9 таблиць з даними. В роботі використано 19 літературних джерел.

Дипломна робота присвячена дослідження та розробка інтернет-магазину на «хмарній» технології.

В розділі «Огляд літературних та інших джерел» проведено огляд основних характеристик, виявлено основні переваги та недоліки «хмарних» обчислень. Виконано аналіз «хмарних» сховищ, розглянуто особливості та переваги кожного «хмарного» сервісу.

В розділі «Аналіз проблеми та постановка задачі дослідження» проаналізовано необхідність розміщення веб-додатків на «хмарному» сервісі та запропоновано методи та алгоритми балансування навантаження у «хмарних» системах.

У розділі «Методи та засоби вирішення проблеми» було проаналізовано підписку надання ресурсів «хмарним» сервісом, спроектовано базу даних на «хмарі», досліджено та розгорнуто веб-сайт в «хмарному» середовищі, використано сервіси моніторингу для веб-додатків, розглянуто балансувальний LoadMaster від Microsoft Azure

В розділі «Спеціальна частина» було представлено веб-додаток у вигляді UML діаграм.

Об'єкт дослідження - розміщення веб-додатку на «хмарному» сервісі.

Мета роботи - розміщення веб-додатка на «хмарному» сервісі. Було проведено аналіз основних помилок веб-додатку. Оцінено доцільність використання балансувальника LoadMaster в «хмарній» системі Microsoft Azure, налаштування балансувальника в Azure Portal.

Предмет дослідження - є використання методів і алгоритмів балансування навантаження веб-додатків у «хмарному» середовищі, критерії їх порівняння, узагальнена модель «хмарного» додатку, обґрунтування вибору правил балансування навантаження відповідно до вимог створюваного сервісу.

Основні результати : проаналізовано веб-сайт на «хмарному» сервісі, виявлено переваги й недоліки веб-сайту на «хмарному» сервісі.

Результати даних досліджень будуть представлені у вигляді діаграм та схем. Нами виявлено в дослідженні використання балансувальника LoadMaster для Microsoft Azure.

Ключові слова: MICROSOFT AZURE, ІНТЕРНЕТ-МАГАЗИН, ХМАРНІ СХОВИЩА, ВЕБ-ДОДАТОК, ХМАРНИЙ СЕРВІС, ХМАРА, LOADMASTER

## ANNOTATION

The thesis consists of an explanatory note and a graphic part (illustrative material - slides). The volume of the graphic part of the thesis is 10 slides. The volume of the explanatory note is 105 A4 printed pages ( $210 \times 297$ ), the volume of the supplement is 14 A4 printed pages. The thesis consists of eight sections, with 47 figures and 9 data tables. 19 literary sources were used.

Everything changes, the world is not standing still, and most Internet users are also changing their attitude to the world wide web. The main reason for this - "cloud technology" that contribute to an increasing use of the Internet, storing files on the web. For the "cloud" technologies are of particular importance scaling and load balancing. Scalability - an important aspect of electronic systems, software systems, database systems, routers, networks, and so on. N. If they need to work under high load. Load balancing directly affects the efficiency of computing, fault tolerance.

To ensure the coordinated work of the computer network nodes on the side of the cloud provider uses specialized middleware software that provides status monitoring and programs, load balancing, providing resources for the task.

Currently information about these aspects "cloud technology" as scaling and load balancing fragmented and requires systematization and ranking. This is the relevance of this work. Also actual problem is the choice of load balancing that meet your goals and criteria it set in accordance with the requirements.

The purpose of the master's thesis is to systematize and compare basic methods and algorithms for load balancing and scaling methods applications in the "cloud" environment, comparison of distributed load balancing of applications research system setup load balancing in CloudStack platform and provide appropriate practical recommendations. To do this, set tasks are solved in the work:

Hold the classification of methods and algorithms for load balancing in the "cloud" systems; make a comparative description of distributed load balancing; provide practical advice on setting system load balancing platform CloudStack an example application of remote video learning. This paper was conducted systematization and comparative description of the basic methods and algorithms for load balancing and scaling methods applications in the "cloud" environment,

conducted comparative description of distributed load balancing applications, system settings investigated load balancing in CloudStack platform and provided such practical recommendations.

The work was completed installation of remote video learning BigBlueButton a virtual machine platform CloudStack, reasonable selection rules and auto-load balancing for an extensive infrastructure and organized setting these facilities. Experimental study of the effectiveness of the rules created balancing, conducted by load testing system confirms the correct choice rules and general system settings.

Holding classification methods and algorithms for load balancing in the "cloud" systems, the development of comparative characteristics common load balancing, provide practical recommendations for setting up the system load balancing platform CloudStack an example application of remote video learning, experimental studies of the effectiveness of load balancing rules created for the selected application.

Scientific novelty lies in the systematization and comparative characterization methods and basic load balancing algorithms and methods for scaling applications in the "cloud" environment, comparative characteristics of distributed load balancing applications research system setup load balancing in CloudStack platform and providing appropriate recommendations for balancing the load in " Cloud Technology "according to the selected criteria.

The work was completed installation of remote video learning BigBlueButton a virtual machine platform CloudStack, reasonable selection rules and auto-load balancing for an extensive infrastructure and organized setting these facilities. Experimental study of the effectiveness of the rules created balancing, conducted by load testing system confirms the correct choice rules and general system settings. The

results of these studies can be used for load balancing technology selection according to specific requirements in courses on "cloud computing" , further work on the introduction of SD private "cloud" service IaaS for distance learning.

Keywords: MICROSOFT AZURE, E-SHOP, CLOUD STORAGE , WEBSITE, CLOUD SERVICES , CLOUD , LOADMASTER

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	8
Вступ.....	9
1. Аналітична частина .....	11
1.1 Огляд технології «хмарних» обчислень.....	11
1.1.1 Принцип роботи «хмар».....	13
1.1.2 Основні характеристики «хмар» .....	13
1.1.3 Переваги та недоліки «хмарних» обчислень .....	15
1.2 Аналіз «хмарних» сховищ.....	17
1.2.1 Google Drive.....	18
1.2.2 OneDrive.....	19
1.2.3. Dropbox .....	20
1.2.4. Яндекс.Диск.....	21
1.2.5. XМара@mail.ru .....	22
1.2.6. Amazon Web Services.....	23
2. Технологічна частина .....	26
2.1 Аналіз необхідних розміщення додатків на «хмарному» сервісі .....	26
2.2 Запропоновані методи і алгоритми балансування навантаження в «хмарних» системах.....	27
2.2.1 Балансування навантаження .....	27
2.2.2 Види балансування навантаження .....	28
2.2.3 Кроки балансування навантаження .....	30
2.2.4 Методи балансування навантаження.....	31
2.2.5 Алгоритми балансування навантаження.....	37
3. Конструкторська частина .....	46
3.1 Підписка надання ресурсів «хмарним» сервісом .....	46
3.2 Проектування бази даних на «хмарі» .....	47
3.3 Дослідження та розгортання веб-сайту в «хмарному» середовищі.....	50
3.4 Сервіси моніторингу для веб-додатків .....	54
3.5 Балансувальник LoadMaster для Microsoft Azure .....	57
3.5.1 Налаштування балансування навантаження в Azure Portal .....	61

4	Науково-дослідна частина.....	67
4.1	Мета, об'єкт, предмет та методи дослідження .....	67
4.2	Результати дослідження .....	68
5	Спеціальна частина.....	70
5.1	Огляд розробленого веб-додатку UML діаграмами.....	70
6	Обґрунтування економічної ефективності.....	77
6.1	Планування товару.....	77
6.2	Розрахунок капітальних витрат на проведення проекту .....	80
6.3	Визначення витрат на оплату праці працівників задіяних у проекті .....	80
6.4	Визначення суми амортизаційних відрахувань .....	82
6.5	Розрахунок вартості електроенергії, необхідної для реалізації проекту .....	83
6.6	Визначення витрат підприємства на сировину і матеріали пов'язані із реалізацією проектного рішення і узагальнення його собівартості .....	84
6.7	Розрахунок вартості програмного продукту .....	85
6.8	Визначення економічної ефективності і терміну окупності капітальних вкладень .....	86
7	Охорона праці та безпека в надзвичайних ситуаціях.....	88
7.1	Охорона праці.....	88
7.1.1	Захист користувачів ПК від різного типу випромінювання.....	88
7.2	Безпека в надзвичайних ситуаціях .....	91
7.2.1	Особливості роботи та розлади здоров'я користувачів комп'ютерів, що формуються під впливом роботи за комп'ютером.....	91
7.2.2	Організації робочого місця користувача відео дисплейним терміналом.....	94
8	Екологія .....	98
8.1	Джерела шуму і вібрацій, методи їх знешкодження .....	98
8.1.1	Вплив шуму на організм людини.....	98
8.1.2	Методи і засоби захисту від шуму .....	99
8.1.3	Дія вібрації на організм людини .....	100
8.1.4	Методи і засоби захисту від вібрації .....	102
	Висновки .....	104
	Перелік посилань.....	106
	Додатки.....	107

## **Перелік умовних позначень, символів, одиниць, скорочень і термінів**

АСУТП – автоматизована система управління технологічними процесами

АСУВ – автоматизована система управління виробництвом

УОМ– управляючі обчислювальні машини

ПЛК – програмований логічний контролер

ПК – персональний комп'ютер

БД – база даних

ПЗ – програмне забезпечення

HTML (HyperText Markup Language) – мова розмітки гіпертексту

DNS (Domain Name System) – система доменних імен

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту

СКБД – система керування базою даних

SQL (Structured query language) – мова структурованих запитів

FTP (File Transfer Protocol) – протокол передачі файлів

GUI (Graphical user interface) – графічний інтерфейс користувача

SNMP (Simple Network Management Protocol) - спрощений протокол керування мережею

WWW (World Wide Web) – всесвітня мережа

див. – дивись

т.ін. – інше



## ВСТУП

Все змінюється, світ не стоїть на місці, і більшість користувачів мережі Інтернет також змінюють своє ставлення до світової павутини. Основна причина тому - «хмарні технології», які сприяють все більшому користуванню Інтернетом, і зберіганню файлів в Мережі. Саме «за хмарою» працюють тепер Amazon, Facebook, Twitter, на яких засновані сервіси типу Google Docs і Gmail.

Термін «хмарні обчислення» (cloud computing) став використовуватися на ринку ІТ з 2008 року. Розробники «хмарних» обчислень визначають їх як інноваційну технологію, вона надає динамічно масштабовані обчислювальні ресурси і програми через Інтернет в якості сервісу під управлінням постачальника послуг.

Про «хмарні» технології та їх новизну потрібно говорити, не про інноваційність «хмарних» технологій, а про перехід кількості в якість, тобто про ефект масштабування. Як приклад цього виду послуг можна навести віртуально виділений сервер - сучасну технологію хостингу, що поєднує в собі потужність виділеного сервера з гнучкістю і простотою управління. Причому в деяких застосуваннях «хмарні» обчислення можуть стати альтернативою суперкомп'ютерів.

Суть «хмарних» обчислень полягає в наданні кінцевим користувачам віддаленого динамічного доступу до послуг, обчислювальних ресурсів і додатків (включаючи операційні системи та інфраструктуру) через Інтернет. Розвиток сфери хостингу був обумовлений тим, що виникла потреба в програмному забезпеченні і цифрових послугах, якими можна було б управляти зсередини, але які були б при цьому більш економічними і ефективними за рахунок економії на масштабі.

Для "хмарних" технологій особливе значення мають масштабування та балансування навантаження. Комп'ютерні технології щодня пропонують все нові і нові варіанти використання персональних серверів і комп'ютерів .

Масштабованість - важливий аспект електронних систем, програмних комплексів, систем баз даних, мереж, маршрутизаторів, якщо для них потрібна можливість працювати під великим навантаженням. В системі з поганою

масштабованістю додавання ресурсів призводить лише до незначного підвищення продуктивності, а з деякого «порогового» моменту додавання ресурсів не дає ніякого корисного ефекту.

Для забезпечення узгодженої роботи вузлів обчислювальної мережі на стороні хмарного провайдера використовується спеціалізоване проміжне програмне забезпечення, що забезпечує моніторинг стану обладнання і програм, балансування навантаження, забезпечення ресурсів для вирішення завдання.

Одним з основних рішень для згладжування нерівномірності навантаження на послуги є розміщення шару серверної віртуалізації між шаром програмних послуг та апаратним забезпеченням. В умовах віртуалізації балансування навантаження може здійснюватися за допомогою програмного розподілу віртуальних серверів по реальним.

На даний момент інформація про такі аспекти "хмарних технологій" як масштабування та балансування навантаження розрізнена та вимагає систематизації та впорядкування. В цьому полягає актуальність даної роботи. Також проблемою є вибір засобу балансування навантаження, що відповідає поставленим задачам та критеріям, його налаштування у відповідності до поставлених вимог.

## 1. Аналітична частина

### 1.1 Огляд технології «хмарних» обчислень

«Хмарні» обчислення (англ. cloud computing), в інформатиці - це модель забезпечення повсюдного і зручного мережевого доступу на вимогу до загального пулу конфігуруючих обчислювальних ресурсів (наприклад, мереж передачі даних, серверів, пристроїв зберігання даних, додатків і сервісів - як разом, так і окремо), які можуть бути оперативно надані і звільнені з мінімальними експлуатаційними витратами або зверненнями до провайдера [1]. Діаграма, що дає уявлення про суть «хмарних» обчислень подана на рисунку 1.1.

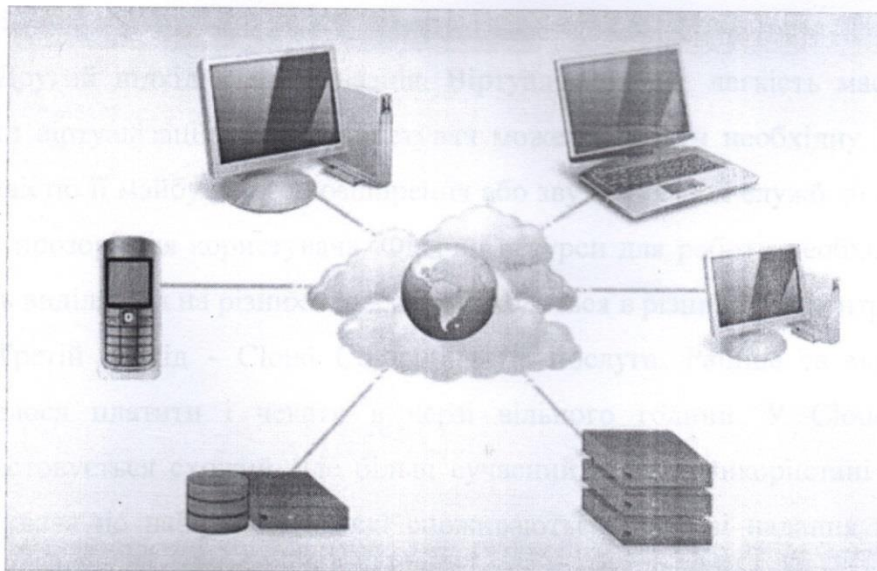


Рисунок 1.1— Схема представлення «хмари»

Коли використовують «хмарні» обчислення програмне забезпечення надається користувачеві як Інтернет-сервіс. Користувач має доступ до власних даних, але не може управляти і не повинен піклуватися про інфраструктуру, операційну систему і програмне забезпечення, з яким він працює. «Хмарою» називають Інтернет, який приховує всі технічні деталі. Згідно з документом IEEE (Institute of Electrical and Electronics Engineers), опублікованим у 2008 році, «Хмарні» обчислення — це парадигма, в рамках якої інформація постійно зберігається на

серверах у мережі Інтернет і тимчасово кешується на клієнтській стороні, наприклад на персональних комп'ютерах, ігрових приставках, смартфонах тощо [2].

В основі Cloud computing лежать кілька підходів.

Перший - доступність сервісів через Інтернет. Цей підхід не відноситься до закритих інфраструктур, в них мережу Інтернет замінюється локальними мережами, але частина сервісів, як правило, все одно доступні з глобальної мережі.

Другий підхід - віртуалізація. Віртуалізація дає легкість масштабування. Завдяки віртуалізації кожен користувач може отримати необхідну потужність з можливістю її майбутнього розширення або звуження. Усі службові процеси, при ньому, прозорі для користувача. Фізичні ресурси для роботи необхідної системи можуть виділятися на різних серверах знаходяться в різних дата центрах.

Третій підхід - Cloud Computing це послуга. Раніше за машинний час доводилося платити і чекати . У Cloud Computing використовується схожий, але більш сучасний підхід. Використані ресурси для користувача це набір послуг, які споживаються і в разі надання комерційним постачальником оплачуються. Наприклад, розглянемо хостінг для даних з доступом до них через HTTP REST API. У користувача є потрібний обсяг даних, такий доступний за допомогою зручного інтерфейсу.

Четвертий підхід - простота і стандартність. Це дуже важлива властивість для нової, ще не повністю адаптованої технології. Все, що пропонується всередині «хмари» доступно через прості виклики API і протоколи.

Величезну популярність знайшов протокол REST, за допомогою якого всі операції над даними можна зробити через запити. Застосовуються й багато інших рішень, для різних мов програмування вже доступні бібліотеки для написання подібних систем роботи з даними.

### **1.1.1 Принцип роботи «хмар»**

Щоб зрозуміти принцип роботи «хмарних» технологій, потрібно уявити, що «пошарована» «хмара» складається, в основному, з «внутрішніх» (back-end) і «користувацьких» (front-end) шарів. Шари front-end - це те, що бачить, те, з чим чає справу користувач. Наприклад, користуючись сервісом Gmail, маємо справу з ЛЗ, працюючим на front-end шарі «хмари». Те ж саме відбувається і коли заходимо під своїм обліковим записом у Facebook . Внутрішні ж шари складаються з власне обладнання та програмних сервісів, що забезпечують роботу того інтерфейсу, що видається на «користувацькому» шарі [1].

Оскільки всі комп'ютери «хмари» налаштовані так, щоб працювати спільно, додаткам доступна вся сумарна потужність цих комп'ютерів, начебто це додаток виконувався на конкретному окремо взятому комп'ютері. «Хмарні» технології мають, крім того, і чималу ступінь гнучкості в роботі. Залежно від потреб, можна підняти чи опустити ту планку використовуваних «хмарою» ресурсів, за якої буде потрібно включення в роботу додаткового обладнання. А можна і зменшити число ресурсів, призначених на роботу, коли вона не є пріоритетною.

### **1.1.2 Основні характеристики «хмар»**

Провайдери «хмарних» рішень дозволяють орендувати через Інтернет обчислювальні потужності та дисковий простір. Переваги такого підходу - доступність (користувач платить за ті ресурси, які йому потрібні) і можливість гнучкого масштабування. Клієнти позбавляються від необхідності створювати і підтримувати власну обчислювальну інфраструктуру. Власної розвиненої ІТ-структури.

"Хмара" відкриває новий підхід до обчислень, при якому ані обладнання, ані програмне забезпечення не належать підприємству. Замість цього провайдер надає замовнику вже готовий сервіс.

До допомоги "хмар" часто вдаються молоді компанії-стартапи, які потребують великих обчислювальних ресурсів для обслуговування користувачів, але не можуть дозволити собі створення і експлуатацію власного дата-центру.

Одним з перших широкодоступних «хмарних» Інтернет-сервісів стала електронна пошта з веб-інтерфейсом. У цьому випадку всі дані зберігаються на віддалених серверах, а користувач отримує доступ до своїх листів через браузер з будь-якого комп'ютера або достатньо потужного мобільного пристрою.

Національним інститутом стандартів і технологій США встановлені такі обов'язкові характеристики «хмарних» обчислень [ 1 ]:

- Універсальний доступ по мережі, послуги доступні споживачам через мережу передачі даних незалежно від термінального пристрою;

- Самообслуговування на вимогу (англ. self service on demand), споживач самостійно визначає і змінює обчислювальні потреби, такі як серверний час, швидкості доступу та обробки даних, обсяг збережених даних без взаємодії з представником постачальника послуг;

- Об'єднання ресурсів (англ. resource pooling), постачальник послуг об'єднує ресурси для обслуговування великої кількості споживачів в єдиний пул для динамічного перерозподілу потужностей між споживачами в умовах постійної зміни попиту на потужності; при цьому споживачі контролюють тільки основні параметри послуги (наприклад, обсяг даних, швидкість доступу), але фактичний розподіл ресурсів, що надаються споживачеві, здійснює постачальник (в деяких випадках споживачі все ж можуть керувати деякими фізичними параметрами перерозподілу, наприклад, вказувати бажаний центр обробки даних з міркувань географічної близькості);

- Еластичність, послуги можуть бути надані, розширені, звужені в будь-який момент часу, без додаткових витрат на взаємодію з постачальником, як правило, в автоматичному режимі;

- Облік споживання, постачальник послуг автоматично обчислює спожиті ресурси на певному рівні абстракції (наприклад, обсяг збережених даних, пропускна здатність, кількість користувачів, кількість транзакцій), і на основі цих даних оцінює обсяг наданих споживачам послуг.

З точки зору споживача, ці характеристики дозволяють отримати послуги з високим рівнем доступності (англ. high availability) і низькими ризиками непрацездатності, забезпечити швидке масштабування обчислювальної системи завдяки еластичності без необхідності створення, обслуговування і модернізації власної апаратної інфраструктури.

З точки зору постачальника, завдяки об'єднанню ресурсів та непостійному характеру споживання з боку споживачів, «хмарні» обчислення дозволяють економити на масштабах, використовуючи менші апаратні ресурси, ніж при виділенні апаратних потужностей для кожного споживача, а за рахунок автоматизації процедур модифікації виділення ресурсів істотно знижуються витрати на абонентське обслуговування.

Зручність і універсальність доступу забезпечується широкою доступністю послуг і підтримкою різного класу термінальних пристроїв.

### **1.1.3 Переваги та недоліки «хмарних» обчислень**

Технології «хмарних» обчислень мають свої переваги та недоліки. Спочатку розглянемо переваги «хмарних» обчислень.

По-перше до них можна віднести доступність. «Хмари» доступні всім, з точки, де є Інтернет, з будь-якого комп'ютера, де є браузер. Це дозволяє користувачам заощаджувати на закупівлі високопродуктивних, дорогих комп'ютерів. Також співробітники компаній стають більш мобільними так, як можуть отримати доступ до свого робочого місця з будь-якої точки земної кулі, використовуючи ноутбук, нетбук, планшет або смартфон. Наступна перевага – це низька вартість. Основні фактори, що знизили вартість використання «хмар», наступні:

-Зниження витрат на обслуговування віртуальної інфраструктури, викликане розвитком технологій віртуалізації, за рахунок чого потрібно менший штат для обслуговування всієї ІТ інфраструктури підприємства;

-Використання «хмари» на правах оренди дозволяє користувачам знизити витрати на закупівлю дорогого устаткування, і зробити акцент на

вкладення грошових коштів на налагодження бізнес процесів підприємства, що в свою чергу дозволяє легко почати бізнес;

-Оплата фактичного використання ресурсів. Користувач платить за фактичне використання обчислювальних потужностей «хмари», що дозволяє йому ефективно розподіляти свої грошові ресурси. Це дозволяє користувачам (підприємствам) заощаджувати на покупці ліцензій до ПЗ;

-Розвиток апаратної частини обчислювальних систем, у зв'язку з чим знижується вартість обладнання.

Також однією з переваг є гнучкість - необмеженість обчислювальних ресурсів (пам'ять, процесор, диски). За рахунок використання систем віртуалізації, процес масштабування і адміністрування «хмар» стає досить легким завданням, скільки «хмара» самостійно може надати ресурси, які необхідні, а плата відбувається тільки за фактичне їх використання.

Ще одна з переваг - надійність. Надійність «хмар», особливо тих, що перебувають у спеціально обладнаних ЦОД, дуже висока так, як такі ЦОД мають резервні джерела живлення, охорону, професійних працівників, регулярне резервування даних, високу пропускну здатність Інтернет каналу, високу стійкість по DoS (Denial of Service) атак.

Важливою перевагою є безпека. «Хмарні» сервіси мають досить високу безпеку при належному її забезпеченні, однак при недбалому ставленні ефект може бути повністю протилежним.

Найважливішою перевагою є великі обчислювальні потужності. Користувач «хмарної» системи може використовувати всі її обчислювальні здібності, заплативши тільки за фактичний час використання.

Підприємства можуть використовувати дану можливість для аналізу великих обсягів даних.

Незважаючи на всі переваги, «хмари» мають і свої недоліки.

Перший з них - це необхідність постійного з'єднання з мережею. Для отримання доступу до послуг «хмари» необхідно постійне з'єднання з мережею Інтернет. Але у наш час це не такий і великий недолік особливо з приходом технологій стільникового зв'язку 3G і 4G.



Наступним недоліком є програмне забезпечення і його кастомізація. Є обмеження на ПЗ яке можна розгортати на «хмарах» і надавати користувачеві. Користувач ПЗ має обмеження у використуваному ПЗ і не має можливості налаштувати його під свої власні цілі.

Конфіденційність даних збережених на публічних «хмарах» в даний час викликає багато суперечок, але в більшості випадків експерти сходяться в тому, що не рекомендується зберігати найбільш цінні для компанії документи на публічній «хмарі», так як в даний час немає технології яка б гарантувала 100% конфіденційність збережених даних.

Що стосується надійності інформації, що зберігається, то з упевненістю можна сказати що якщо було втрачено інформацію збережену в «хмарі», то її втрачено назавжди. Це безперечно недолік.

Безпека є також недоліком. «Хмара» сама по собі є достатньо надійною системою, однак при проникненні на неї зловмисник отримує доступ до величезного сховища даних. Також є використання систем віртуалізації, в яких використовуються ядра стандартних ОС таких, як Linux Windows , що дозволяє розповсюджувати віруси програми.

Якщо розглядати економічні показники використання «хмарних» технологій, то можна виділити ще один недолік - дороговизна обладнання. Щоб збудувати власну «хмару» компанії необхідно виділити значні матеріальні ресурси, що не вигідно щойно створеним і малим компаніям.

## **1.2 Аналіз «хмарних» сховищ**

«Хмарне» сховище даних - модель онлайн-сховища, в якому дані зберігаються на численних, розподілених в мережі серверах, що надаються в користування клієнтам, в основному третьою стороною. На противагу моделі зберігання даних на власних, виділених серверах, придбаних або орендованих спеціально для подібних цілей, кількість або будь-яка внутрішня структура серверів клієнту. Дані зберігаються і обробляються, в так званій «хмарі», яке представляє собою, один великий, віртуальний сервер.

Розглянемо та проаналізуємо безкоштовні та платні сервіси «хмарних» сховищ даних та проведемо аналіз їх послуг.

### 1.2.1 Google Drive

Google Drive – це один з найпопулярніших «хмарних» сховищ даних, що дозволяє користувачам зберігати свої дані на серверах в «хмарі» і ділитися ними з іншими користувачами в інтернеті. Після активації, користувач для роботи з документами, трансформували його в «хмарний» сервіс і додали безкоштовного простору. У сервісі можна зберігати не тільки документи, але і фотографії, музику, відео та багато інших файли - всього близько 30 типів. Але взагалі все дуже зручно і звично для користувачів Google-сервісів (див. рис. 1.2.1).

Кожному користувачеві Google Drive надається безкоштовно 15Gb простору. Якщо виділеного обсягу недостатньо, можна придбати додатково до 30 ТБ:

Обсяг Місячна плата:

- 100 GB\$ 1.99
- 1 TB \$ 9.99
- 10 TB\$ 99.99
- 20 TB \$ 199.99
- 30TBS 299.99



Рисунок 1.2.1 - Google Drive

Крім доступу до сервісу через веб-інтерфейс, є можливість доступу через клієнти для Windows, Mac OS і Android, iOS.

## 1.2.2 OneDrive

OneDrive - це онлайнове сховище, яке отримується разом із обліковим записом Microsoft. Зберігаючи дані в сховищі, їх можливо отримати із будь-якого комп'ютера, планшета або телефону.



Рисунок 1.2.2 – OneDrive

Перейменований в лютому 2014 Microsoft SkyDrive, інтернет-сервіс зберігання файлів з функціями файлообміну. SkyDrive створений в серпні 2007 року компанією Microsoft. Зараз OneDrive один з флагманів «хмарних» сховищ даних.

Перевагою сервісу OneDrive в тому, що він відразу інтегрований з Office 365, тому безпосередньо з програми можна створювати, редагувати та зберігати файли Excel, OneNote, PowerPoint і Word в службі Windows Live OneDrive.

Сервіс OneDrive зберігає на даний момент безкоштовно 5 Гбайт інформації в упорядкованому за допомогою стандартних папок вигляді. Для зображень передбачений попередній у вигляді ескізів, а також можливість їх перегляду в слайдах.

OneDrive надає такі пакети:

- Free: 5 Гбайт - безкоштовно;
- Basic: 50 Гбайт - \$ 1,99 на місяць;
- Personal: 1 ТВ для 1 користувача - \$ 6,99 на місяць;
- Home: 1 ТВ для кожного з 5 користувачів - \$ 69,99 в місяць

### 1.2.3. Dropbox

Dropbox – «хмарне» сховище даних, що дозволяє користувачам зберігати свої дані на серверах в хмарі і розділяти їх з іншими користувачами в інтернеті. Його робота побудована на синхронізації даних.



Рисунок 1.2.3 – Dropbox

Тарифні плани Dropbox:

-Базовий аккаунт Dropbox безкоштовно 2 ГБ; Надійне резервне копіювання, доступ звідусіль, простий спільний доступ;

-Розширений - € 9.99 в місяць 1 ТБ; Всі функції Базового аккаунта Dropbox , а також: додатковий контроль доступу, віддалене стирання даних;  
Dropbox для Бізнесу - 12 €/користувач/місяць.

Складається враження, що 2 Гб дискового простору замало, у Dropbox є цілий ряд безкоштовних способів, крім платних, за допомогою яких можна збільшити цей простір з 2 Гбайт до 48 Гбайт:

-500 Мбайт за запрошеного друга (до 16 Гбайт);

-для того, щоб отримати 48 Гбайт безкоштовного дискового простору, потрібно виконати 7 кроків:

-переглянути огляд про Dropbox;

-встановити програму Dropbox на свій комп'ютер;

-покласти файли в папку Dropbox;

- встановити Dropbox на інші комп'ютери, які ви використовуєте;
- надати загальний доступ друзям і колегам;
- запросіть своїх друзів в Dropbox;
- встановіть програму Dropbox на ваш мобільний пристрій;

На відміну від основних конкурентів, при роботі з Dropbox редаговані файли не копіюються повністю на сервер - здійснюється передача тільки зміненої частини, попередньо стиснутої. Вважається, що саме цей факт багато в чому пояснює відому оперативність роботи з Dropbox, в порівнянні з аналогами.

Головним недоліком Dropbox можна вважати підхід до вибору папок для лінхронізації. Фактично додаток стежить за вмістом тільки однієї папки - Dropbox.

Не всі фахівці з безпеки згодні з такою заявою. Крім того, використання Dropbox у зв'язці з Dropbox, який надійно шифрує файли перед синхронізацією їх з «хмарою», забезпечує конфіденційність даних в Dropbox.

Ще один важливий недолік - це те, що додаткові гігабайти на Dropbox, як правило, мають термін придатності. І може так статися, що в якийсь момент, ваш Dropbox стане переповненим, тому що всі квоти закінчуються.

#### **1.2.4. Яндекс.Диск**

Яндекс.Диск - безкоштовний «хмарний» сервіс від Яндекса, що дозволяє користувачам зберігати свої дані на серверах в «хмарі» і передавати їх іншим користувачам в інтернеті. Робота побудована на синхронізації даних між різними пристроями. Зареєстрованому користувачу Яндекс.Диск надає близько 10Гб назавжди.

Послуги:

Платні пакети хмарного сервісу Яндекс.Диск:

-10 Гбайт - 30 руб. / Міс

-100 Гбайт - 80 руб. / Міс

-1 Тб - 200 руб. / Міс

Бонусне додаткове місце:

-До 10 Гбайт за приведених друзів (по 512 Мб за кожного друга)

-До 50 Гбайт за акції з партнерами - вони змінюються, потрібно стежити за новинами.



Рисунок 1.2.4-Яндекс.Диск

Крім того, Яндекс.Диск може виступати в якості служби «хмарного» сервісу, інтегруючись в офісний пакет Microsoft Office 2013, а недавно з'явилася можливість автоматичного завантаження фото та відео файлів з цифрових камер і зовнішніх носіїв інформації на Яндекс.Диск. При цьому користувачу надаються додатково 32 ГБ простору на півроку.

### 1.2.5. Хмара@mail.ru

Хмара@mail.ru - це хмарне сховище даних від компанії Mail.Ru Group, що і дозволяє користувачам зберігати свої дані в хмарі і синхронізувати дані на різних пристроях, а також ділитися ними з іншими користувачами. «Хмарне» сховище Хмара@mail.ru містить великий розмір дискового простору, що надається безкоштовно. Користувачі можуть відразу безкоштовно отримати 25 Гбайт гарного сховища.

Користуватися хмарою можна не тільки через веб-інтерфейс, але через десктопні і мобільні додатки для Android і iOS.



Рисунок 1.2.5-Хмара@mail.ru

### 1.2.6. Amazon Web Services

Amazon Web Services (AWS) - це цілий набір глобальних сервісів. Включає в себе широкий спектр фундаментальних сервісів «хмарної» інфраструктури.

Зберігання даних і доставка контенту:

- об'єктне сховище;
- CDN;
- блочне сховище;
- сховище файлових систем Обчислення;
- Віртуальні сервери;
- Керовані обчислювальні функції;
- Load Balancing Бази даних;
- реляційні БД;
- Міграція баз даних;
- сховище даних Мережеві рішення;
- Virtual Private Cloud;
- прямі підключення;

- балансування навантаження.



Рисунок 1.2.6 - Amazon Web Services

Крім того, AWS має багатий вибір сервісів для підвищення ефективності вашої «хмари»: аналітика, корпоративні програми, мобільні сервіси, дозволяє легко створювати додатки «Інтернету речей» для збору, обробки, аналізу і зміни даних, що генеруються підключеними пристроями, без необхідності керувати будь-якою інфраструктурою.

Вартість AWS гнучка, це означає, що ви платите тільки за те, чим користуєтесь. Мінімальний збір не стягується. Оцінити свій щомісячний рахунок можна за допомогою калькулятор матеріалів AWS (ціни залежать від вашого місця розташування)/

### **Висновок до 1 розділу**

Отже, «хмарні обчислення», в інформатиці - це модель забезпечення щоденного і зручного мережевого доступу на вимогу до загального пулу конфігуруючих обчислювальних ресурсів, які можуть бути надані і звільнені з мінімальними експлуатаційними витратами або зверненнями до провайдера.

«Хмари» мають свої переваги та недоліки. До переваг можна віднести:



- доступність з будь-якої місця;
- економія коштів користувача на власній інфраструктурі;
- доступність для користувача великих обчислювальних потужностей;
- легкість процесу масштабування і адміністрування «хмари» ;
- надійність апаратного забезпечення «хмари»;
- економія на покупці ПК

Серед недоліків «хмарних» обчислень:

- необхідність постійного з'єднання з мережею;
- велика пропускна здатність з'єднаної мережі;
- проблеми інформаційної безпеки;
- висока вартість побудови «хмари».

## 2. Технологічна частина

### 2.1 Аналіз необхідних розміщення додатків на «хмарному» сервісі

Вказані технології працюють наступним чином: замість придбання, встановлення та керування власними серверами для запуску додатків, забезпечують оренду серверу у компанії. Така компанія надає послуги на основі «хмарних» технологій (cloud- провайдера). Надалі користувачі керують серверами через Інтернет, оплачуючи при цьому тільки фактичне їх використання для обробки та зберігання даних. Більшість сервіс-провайдерів пропонують «хмарні» сервіси в формі VPS-хостингу (хостинг на основі віртуальних приватних серверів), віртуального хостингу і ПЗ як послуги (SaaS) . «Хмари» відкрили перед різними компаніями та організаціями технології, раніше доступні тільки великим корпораціям. Можливість орендувати -Т-рішення, а не платити за ліцензії, звільнити себе від турбот по технічному убезпеченню - все це дозволяє організаціям контролювати свої ІТ-витрати і приділяти більше уваги основній сфері діяльності. Важлива перевага - можливість вибору іншого сервіс-провайдера, або повна відмова від послуги . Для демонстрації різниці між звичайними і «хмарними» сервісами можна користуватись послугами, що надаються відповідним провайдером. При традиційному (звичайному) підході провайдер на місячній основі отримує фіксовану плату за використання його обчислювальних ресурсів ( CPU, RAM, HDD ) . Не має різниці, використовував клієнт виділені йому ресурси в повному обсязі протягом усього місяця чи тільки декілька днів, а решту часу обчислювальні ресурси простоювали. Надання послуг «хмарного» сервісу, як правило використовують тип оплати «плата за використання». За одиницю виміру часу роботи приймається хвилина чи година користування ресурсами. При оцінці обсягів даних за одиницю виміру приймається мегабайт. Користувач оплачує рівно той обсяг ресурсів, який їм реально використовувався протягом певного часу. «Хмарна» інфраструктура надає користувачеві можливість при необхідності збільшувати або зменшувати ліміти виділених ресурсів, користувачеві «хмарних» сервісів немає необхідності піклуватися про інфраструктуру, яка забезпечує роботу сервісів. Завдання з

налаштування, усунення несправностей, розширення інфраструктури, тощо бере на себе сервіс- провайдер. «Хмарні» середовища можна розділити на два типи: публічні і приватні (private cloud). Приватна «хмара» володіє багатьма перевагами комп'ютерного середовища на базі публічних «хмар» але в ній управління даними і процесами здійснюється всередині організації. Іншими словами, в цьому випадку відсутні такі проблеми, як обмеження пропускну здатності мережі та загрози безпеки, які могли б виникнути при використанні публічних хмар за допомогою відкритих мереж загального користування. На сьогодні визначено три рівні «хмарних» сервісів. Інфраструктура як сервіс (IaaS). Інфраструктура в оренду. Користувачеві вдається «чистий» віртуальний сервер з унікальною IP-адресою. Використовується частина системи зберігання даних. Для управління параметрами цього сервера провайдер надає користувачеві програмний інтерфейс (API). Платформа як сервіс (PaaS). PaaS можна представити як готову до роботи віртуальну платформу, що складається з одного або декількох віртуальних серверів з встановленими системами і спеціалізованими додатками. Більшість «хмарних» провайдерів пропонують користувачеві вибір з маси готових до використання середовищ. Програмне забезпечення як сервіс. Підхід SaaS надає можливість користуватися програмним забезпеченням як послугою і робити це віддалено через Інтернет. Даний підхід дозволяє не купувати "програмний продукт, а на певний час використовувати його при виникненні потреби. Перевага «хмарних» послуг - швидке надання їх та доступ до ресурсів в будь-якому місці і в будь-який час.

## **2.2 Запропоновані методи і алгоритми балансування навантаження в «хмарних» системах**

### **2.2.1 Балансування навантаження**

У термінології комп'ютерних мереж, балансування навантаження, або вирівнювання навантаження (англ. Load balancing) - метод розподілу завдань між кількома мережевими пристроями (наприклад, серверами) з метою оптимізації використання ресурсів, скорочення часу обслуговування запитів,

горизонтального масштабування кластера (динамічне додавання), а також забезпечення відмовостійкості [7].

Балансування навантаження може бути використане для розширення можливостей ферми серверів, що складається більш ніж з одного сервера. Дане навантаження може дозволити продовжувати роботу навіть в умовах, коли кілька виконавчих пристроїв (серверів) вийшли з ладу. Завдяки цьому зростає відмовостійкість, і з'являється можливість динамічно регулювати використовувані та обчислювальні ресурси, за рахунок додавання/видалення виконавчих пристроїв в кластері.

Балансування навантаження передбачає рівномірне навантаження обчислювальних вузлів. Завдяки появі нових завдань програмне забезпечення, що реалізує балансування, має прийняти рішення про те, де слід виконувати обчислення, пов'язані з цим новим завданням. Також, балансування передбачає перенос (migration - міграція) частини обчислень з найбільш навантажених обчислювальних вузлів на менш завантажені вузли.

Проблема балансування обчислювального навантаження виникає з тієї причини, що:

- структура обчислювального комплексу (наприклад, кластера), також - однорідна, тобто різні обчислювальні вузли володіють різною продуктивністю;
- структура розподіленого додатка неоднорідна, різні логічні процеси вимагають різних обчислювальних потужностей;
- структура між вузлової взаємодії неоднорідна, тому лінії зв'язку, що з'єднують вузли, можуть мати різні характеристики пропускну здатності.

### **2.2.2 Види балансування навантаження**

Розрізняють статичне і динамічне балансування. Статичне балансування виконується до початку виконання розподіленого додатка. Часто при розподілі логічних процесів по процесорах використовується досвід попередніх виконань, застосовуються генетичні алгоритми[8]. Однак попереднє розміщення логічних процесів по процесорах (комп'ютерів) не дає ефекту.

Це пояснюється тим, що:

-Може змінитися обчислювальна середовище, в якому відбувається виконання програми, який або обчислювальний вузол може вийти з ладу.

-Обчислювальний вузол, на якому виконується розподілене додаток, зайнятий іншими обчисленнями, частка яких з часом може зрости.

Так чи інакше, виграш від розподілу логічних процесів з комп'ютерів з метою виконання паралельної обробки стає неефективним.

Динамічне балансування передбачає перерозподіл обчислювального навантаження на вузли під час виконання програми. Програмне забезпечення, яке реалізує динамічне балансування, визначає:

- завантаження обчислювальних вузлів;
- пропускну спроможність ліній зв'язку;
- частоту обмінів повідомленнями між логічними процесами розподіленого додатка.

На підставі зібраних даних приймається рішення про перенесення логічних процесів з одного вузла на інший.

Даний процес можна представити на графіку залежності числа оброблених запитів від кількості надійшовших до системи, що складається з двох вузлів А і Б. При оптимальному розподілі навантаження -  $1/3$  запитів на А і  $2/3$  запитів на Б (на рисунку 2.1.а) - величина граничного сумарного числа оброблених запитів, при якому не відбувається скидання, удвічі вище, ніж для варіанту (див. рис 2.1.б). Причина - швидке досягнення межі можливостей вузла А і деградація продуктивності всієї системи [9].

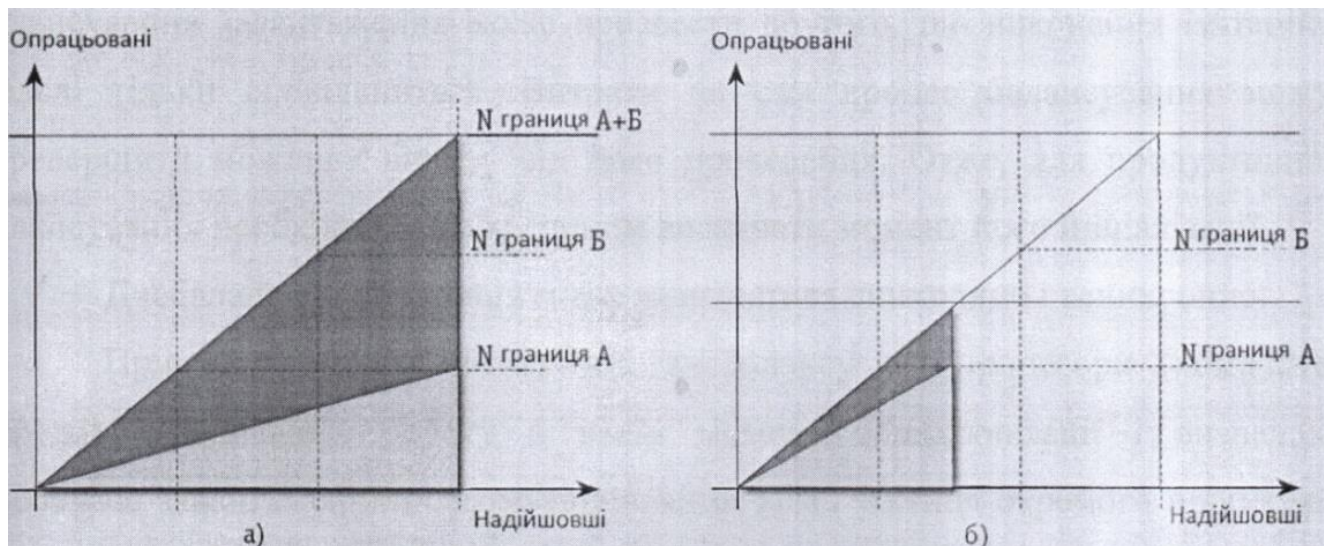


Рисунок 2.1 - Залежність числа оброблених запитів від числа надійшовших запитів

Щоб досягнути рівномірного розподілу обчислювального навантаження, необхідно враховувати насамперед поточне завантаження центрального процесора і ряд характеристик, їх можуть використовувати як індикатори, що вказують на вихід апаратної платформи з допустимого режиму оптимального функціонування.

### 2.2.3 Кроки балансування навантаження

Здебільшого, практичне і повне рішення задачі балансування завантаження складається з чотирьох кроків[10]:

-Оцінка завантаження обчислювальних вузлів. На цьому етапі здійснюється приблизна оцінка завантаження кожного процесора. Інформація про завантаження використовується в якості бази даних для процесу збалансування, по-перше, для визначення виникнення дисбалансу, по-друге, для визначення нового розподілу об'єктів імітаційної моделі шляхом обчислення обсягу робіт, необхідного для переміщення об'єктів.

Якість роботи балансування завантаження безпосередньо залежить від точності і повноти інформації в базі даних.

-Ініціація балансування завантаження. Занадто часте виконання балансування завантаження ,може призвести до того, що виконання імітаційної моделі тільки сповільниться. Витрати на сам процес балансування можуть перевершити можливу вигоду від його проведення. Отже, для продуктивності балансування необхідно якимось чином визначати момент його ініціалізації.

Дисбаланс завантаження може визначатися синхронно і асинхронно.

При синхронному визначенні дисбалансу всі процесори переривають роботу в певні моменти синхронізації і визначають дисбаланс завантаження шляхом порівняння завантаження окремого процесора з загальної середньої завантаженням.

Завдяки асинхронному визначенні дисбалансу кожен процесор зберігає історію своєї завантаження. У цьому випадку момент синхронізації для визначення ступеня дисбалансу відсутня. Обчисленням обсягу дисбалансу займається фоновий процес, що працює паралельно з додатком.

-Прийняття рішень про балансування. Більшість стратегій динамічного балансування завантаження можна віднести до класу централізованих або до класу повністю розподілених.

При централізованій стратегії спеціальний комп'ютер збирає глобальну інформацію про стан всієї обчислювальної системи і приймає рішення про переміщення завдань для кожного з комп'ютерів.

-Переміщення об'єктів. Після прийняття рішень про балансування відбувається переміщення об'єктів серед процесорів для досягнення нового балансу завантаження. При переміщенні об'єкта повинна забезпечуватися цілісність його стану.

Для "хмарних" обчислень останній крок заміняється на переадресацію запитів, які надходять до системи на екземпляр, який визначено як підходящий.

#### **2.2.4 Методи балансування навантаження**

Для прикладу , ми створили кілька серверів (будь-якого призначення - http, база даних), кожен з яких може обробляти запити.

Робота з сайтом переважно не обмежується одним запитом. Тому при проектуванні потрібно зрозуміти, чи можуть послідовні запити клієнта бути коректно оброблені різними серверами, або клієнт повинен бути прив'язаний до одного сервера на час роботи з сайтом. Це дуже важливо, якщо на сайті зберігається тимчасова інформація про сесію роботи користувача (у такому випадку теж можливий вільний розподіл – тоді необхідно зберігати сесії в загальному для всіх серверів сховище). «Прив'язати» відвідувача до конкретного сервера можна за його IP-адресою (яка, може мінятися), або по cookie (в яку заздалегідь записаний ідентифікатор сервера).

Процедура балансування здійснюється за допомогою цілого комплексу алгоритмів і методів, відповідним наступним рівням моделі OSI [11]:

- мережевому;
- транспортному;
- прикладному.

Розглянемо ці рівні більш глибоко.

Балансування на мережевому рівні передбачає вирішення наступного завдання: потрібно зробити так, щоб за одну конкретну IP-адресу сервера відповідали різні фізичні машини. Таке балансування може здійснюватися за допомогою декількох способів.

-DNS-балансування. На одне доменне ім'я виділяється кілька IP-адрес. Сервер, на який буде направлений клієнтський запит, зазвичай визначається за допомогою алгоритму Round Robin (про методи і алгоритмах балансування буде детально розказано нижче).

-Побудова NLB-кластеру. При використанні цього способу сервери об'єднуються в кластер, що складається з вхідних і обчислювальних вузлів. Використовується в рішеннях від компанії Microsoft.

-Балансування за IP з використанням маршрутизатора.

-Балансування за територіальною ознакою здійснюється шляхом розміщення однакових сервісів з однаковими адресами в територіально різних регіонах Інтернету.



Переваги балансування на третьому рівні:

- незалежність від протоколу високого рівня;
- абсолютна прозорість для серверів;
- незалежність від мережевого розташування серверів.

До недоліків:

- весь трафік серверів повинен проходити через балансування , тому він повинен витримувати високе навантаження;
- правила розподілу є статичними, вони не аналізують доступність і поточну завантаженість серверів;
- не підтримує довгострокових сесій між користувачем і сервером.

Балансування на транспортному полягає в тому, що клієнт звертається до балансувальника, який перенаправляє запит одному з серверів і буде його обробляти. Вибір сервера, на якому обробляється запит, може здійснюватися відповідно до самих різних алгоритмів (про це піде мова нижче): шляхом простого кругового перебору, шляхом вибору найменш завантаженого сервера з пулу. На транспортному рівні спілкування з клієнтом замикається на балансувальник, який працює як проксі. Він взаємодіє з серверами від свого імені, передаючи інформацію про клієнта в додаткових даних і заголовках. Таким чином працює, наприклад, популярний програмний балансувальник NARoxy.

До переваг балансування на транспортному рівні можна віднести можливість балансування навантаження незалежно від типу протоколу прикладного рівня до будь-яких TCP-сервісів: HTTPS, SMTP, SSH, FTP, SQL.

При балансуванні на прикладному рівні працює в режимі «розумного проксі». Він аналізує клієнтські запити і перенаправляє їх на різні сервери залежно від характеру запитуваного контенту. В такому разі у проксі- сервера може виникнути потреба в переформуванні адрес кожної веб-сторінки (перетворення імен ресурсів доступних зовні, в імена доступні у внутрішній

мережі), якщо HTTP протоколу треба передати IP-адресу клієнта, то можна додати заголовок X-Real-IP з IP-адресою клієнта. Так працює, наприклад, веб-сервер Nginx, розподіляючи запити між фронтенда і бекенд. За балансування в Nginx відповідає модуль Upstream.

Спеціалізовані апаратно-програмні рішення, які виконують балансування з використанням трансляції заголовків мережевого, транспортного та рівня додатків, завдяки високій продуктивності мають подолати відповідний недолік балансування з пропуском трафіку через один пристрій. Вони підтримують контроль за встановленими сесіями на окремі сервери, постійно контролюють час відповіді серверів, навантаження на які вони розподіляють, і аналізують кількість запитів, які були їм відправлені. Додатково на ці пристрої можливе винесення функцій формування захищеного з'єднання SSL і стиснення даних, що дозволяє значно знизити навантаження на сервери додатків.

Також, можна виділити наступні класи рішень, використовувані при побудові багатовузлових систем (на рисунку 2.2):

-Балансування з пропуском трафіку через один пристрій балансування. Клієнт надсилає запит на один фіксований, відомий йому сервер, а той перенаправляє запит на один з робочих серверів. Типовий приклад - система з одним frontend і декількома backend-серверами, на які відправляються запити. Однак «клієнт» може знаходитися і всередині нашої системи - наприклад, скрипт може слати запит до проксі-сервера бази даних, який передасть запит одному з серверів СУБД. Сам балансуєчий вузол може працювати як на окремому сервері, так і на одному з робочих серверів. Переваги цього підходу в тому, що клієнту нічого не потрібно знати про внутрішній устрій системи—про кількість серверів, про їх адреси та особливості

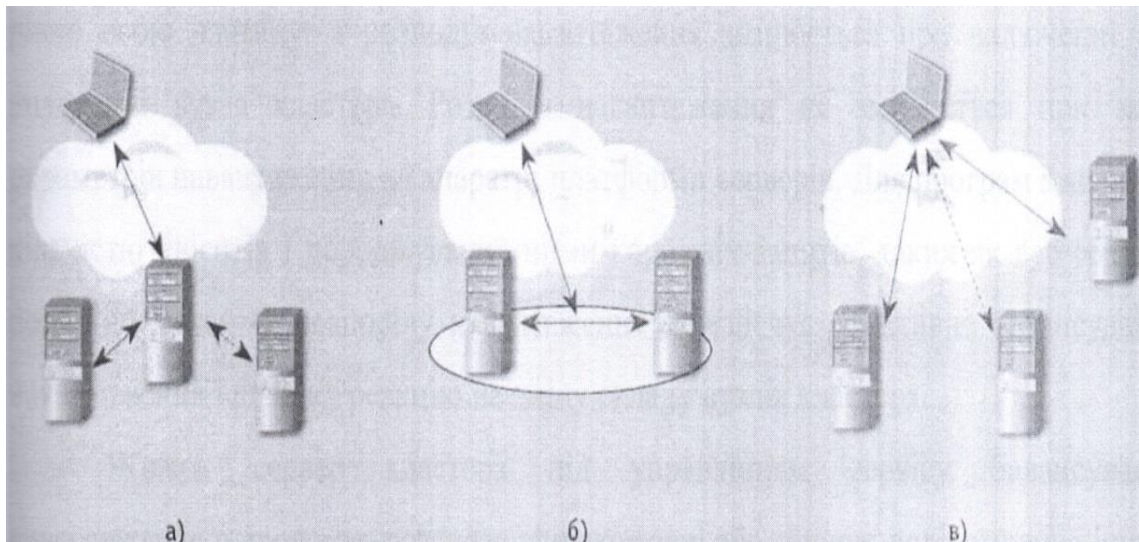


Рисунок 2.2 - Принципи балансування навантаження: а) балансування з пропуском трафіку через один пристрій балансування; б) балансування засобами кластера; в) балансування без пропуску трафіку через один пристрій балансування

всю цю інформацію знає тільки балансувальник. Однак недолік в тому, що балансувальний вузол є єдиною точкою відмови системи - якщо він вийде з ладу, вся система виявиться непрацездатна. Крім того, при великому навантаженні балансувальник може просто перестати справлятися зі своєю роботою, тому такий підхід не завжди застосований. Виходом з цієї ситуації є постійне резервування та перевірка стану машин.

-Балансування засобами кластера. Кластеризація дозволяє управляти групою незалежних серверів як єдиною системою, що підвищує відмовостійкість, спрощує управління і дозволяє домогтися більшої масштабованості. Сервіс балансування в кластері забезпечує розподіл потоку IP- даних між вузлами. У кластері кілька апаратних платформ найчастіше розділяють єдину IP-адресу.

Для кожного вузла, який бере участь у розподілі навантаження, адміністратор вказує його конкретну частку навантаження, або за замовчуванням навантаження буде рівномірно розподілятися між вузлами. Запити клієнтів статистично розподіляються між вузлами, щоб кожен сервер обробляв рівно свою частину, а розподіл навантаження змінюється при включенні або

видаленні вузла кластера. Розподіл навантаження не змінюється при зміні параметрів навантаження на апаратні платформи серверів. Для програм з великою кількістю клієнтів і породжуваних ними коротких запитів, таких як веб-сервіси, подібний механізм розподілу навантаження забезпечує ефективне балансування навантаження і швидку реакцію на зміну складу вузлів кластера.

Кожен сервер кластера під управлінням сервісу балансування навантаження періодично розсилає широкомовні або групові повідомлення іншим вузлам та приймає такі ж повідомлення від інших членів кластеру. Якщо є збої у сервері, решта вузли перерозподіляють навантаження, забезпечуючи безперервне обслуговування. Незважаючи на те, що дані існуючих підключень до втраченого вузла губляться, сервіси залишаються доступними. У більшості випадків, як, наприклад, при балансуванні навантаження веб-серверів, браузер на стороні клієнта автоматично повторює підключення, що закінчилися збоєм, і це відбивається на клієнті тільки у вигляді короткої затримки відгуку на запит. Подібним чином працюють механізми балансування мережного навантаження в кластерах Windows Server 2008 R2, Oracle Solaris Cluster, Open HA Cluster і Linux Virtual Server.

До недоліків рішення на базі кластерів слід віднести необхідність розташування серверних платформ в одному сегменті мережі, неможливість забезпечити мультиплатформенний кластер, необхідність використання на кожному вузлі спеціалізованого ПЗ для функціонування кластера, в завдання якого входить контроль стану суміжних вузлів в кластері і диспетчеризація надходять запитів.

-Балансування без пропуску трафіку через один пристрій балансування. Якщо ми хочемо уникнути єдиної точки відмови, існує альтернативний варіант - доручити вибір сервера самому клієнтові. У такому випадку клієнт повинен знати про внутрішній устрій нашої системи, щоб уміти правильно вибирати, до якого сервера звертатися. Безсумнівним плюсом є відсутність точки відмови - при відмові одного з серверів клієнт зможе звернутися до інших.

Однак платою за це є ускладнення логіки клієнта і менша гнучкість балансування.

Даний варіант балансування організовується шляхом відправки первісного запиту від користувача (наприклад, запиту IP-адреси для встановлення з'єднання або першого запиту на встановлення HTTP-сесії) на виділений сервер- балансувальник. Після отримання запиту балансувальник вказує, з яким сервером додатків з інфраструктури «хмари» продовжувати роботу (повертаючи відповідну IP-адресу або перенаправляючи його засобами HTTP Redirect). Подальша взаємодія відбувається без участі балансувальника[9].

### **2.2.5 Алгоритми балансування навантаження**

Існує безліч різних алгоритмів балансування навантаження. Вибираючи конкретний алгоритм, потрібно виходити, по-перше, із специфіки конкретного проекту, а по-друге - з цілей, які ми плануємо досягти.

Прискорення виконання завдань користувачів на ресурсах, саме це і є основною метою алгоритмів балансування навантаження.

Алгоритм динамічного балансування навантаження має в певний час оцінювати інформацію про стан обчислювального ресурсу.

Є два спекти балансування навантаження в розподілених середовищах: будь-які обсяги обчислень для різних типів завдань і неоднорідність ресурсів , неоднакову продуктивність процесорів.

У числі цілей, для досягнення яких використовується балансування, потрібно виділити наступні [11]:

-справедливість: потрібно гарантувати, щоб на обробку кожного запиту виділялися системні ресурси і не допустити виникнення ситуацій, коли один запит обробляється, а всі інші чекають своєї черги;

-скорочення часу виконання запиту: необхідно забезпечити мінімальний час між початком обробки запиту (або його постановкою в чергу на обробку) і його завершення;

-ефективність: всі сервери, які обробляють запити, повинні бути зайняті на 100%; бажано не допускати ситуації, коли один з серверів простоює в очікуванні запитів на обробку (відразу ж обмовимося, що в реальній практиці ця мета досягається далеко не завжди);

-скорочення часу відгуку: потрібно мінімізувати час відповіді на запит користувача.

Бажано також, щоб алгоритм балансування володів такими властивостями:

-передбачуваність: необхідно чітко розуміти, в яких ситуаціях і за яких навантажень алгоритм буде ефективним для вирішення поставлених завдань;

-рівномірне завантаження ресурсів системи;

-масштабованість: алгоритм повинен зберігати працездатність при збільшенні навантаження.

Round Robin, або алгоритм кругового обслуговування, являє собою перебір по круговому циклу: перший запит передається одному серверу, потім наступний запит передається іншому і так до досягнення останнього сервера, а потім все починається спочатку. Найпоширенішою імплементацією цього алгоритму є, звичайно ж, метод балансування Round Robin DNS.

Round robin DNS - один з методів розподілу навантаження, або відмовостійкості за рахунок надмірності кількості серверів, за допомогою управління відповідями DNS-сервера відповідно до якоїсь статистичної моделі. Звичайно застосовується до таких інтернет-протоколів, як веб-сервери, FTP-сервери[12].

У найпростішому випадку Round robin DNS працює, відповідаючи на запити не тільки одною IP-адресою, а списком з декількох адрес серверів, що надають ідентичний сервіс. Порядок, в якому повертаються IP-адреси зі списку, заснований на алгоритмі round-robin. З кожною відповіддю послідовність ip-адрес змінюється. Прості клієнти намагаються встановлювати з'єднання з першою адресою зі списку, таким чином різним клієнтам будуть з задані адреси різних серверів, що розподілить загальне навантаження між серверами.

Не існує стандартної процедури для визначення того, які адреси будуть використовуватися - деякі сервери намагаються змінити порядок списку, приділяючи пріоритетну увагу чисельно більш «близьким» мережам. Деякі - настільні клієнти намагаються отримати альтернативні адреси після того, як не здалося встановити з'єднання протягом 30-45 секунд.

Кругова система DNS часто використовується для розподілу навантаження територіально розподілених веб-серверів. Наприклад, у компанії є один домен і три ідентичних веб-сайти, розташованих на трьох серверах з трьома різними адресами. Коли один користувач отримує доступ до головної сторінки, він буде направлений на першу адресу IP. Другий користувач, що звертається до головної сторінки, буде відправлений на наступну адресу IP, а третій користувач буде відправлений на третю адресу. У кожному випадку, коли IP-адреса видається, вона відправляється в кінець списку. Четвертий користувач буде відправлений знову на першу адресу IP, і так далі.

Хоча Round robin DNS (RR DNS) легко реалізувати, все ж цей алгоритм має кілька проблематичних недоліків, пов'язаних з кешуванням записів в ієрархії RR DNS самого себе, а також з кешуванням на стороні клієнта, виданої адреси та її повторного використання, поєднання яких важко керовано. RR DNS не спирається на доступність послуг. Для прикладу, якщо сервіс на одній з адрес недоступний, RR DNS буде продовжувати роздавати цю адресу, клієнти будуть як і раніше намагатися з'єднатися з непрацюючим сервером.

Кругова система DNS навантаження найкраще підходить для послуг з великою кількістю рівномірно розподілених з'єднань з серверами еквівалентної потужності. В іншому випадку він просто робить розподіл навантаження.

Існують методи, щоб подолати такі обмеження. Наприклад, модифіковані DNS-сервера можуть регулярно опитувати дзеркала серверів для перевірки їх доступності та завантаженості. Якщо сервер не відповідає в міру необхідності, сервер може бути тимчасово вилучений з пулу DNS, поки він не повідомить, що знову працює у відповідності зі специфікацією.

У числі безперечних плюсів цього алгоритму слід назвати, по-перше, незалежність від протоколу високого рівня. Для роботи за алгоритмом Round

Robin використовується будь-який протокол, в якому звернення до сервера йде по імені.

Балансування на основі алгоритму Round Robin ніяк не залежить від навантаження на сервер: кешуючі DNS-сервери допоможуть впоратися з будь-яким напливом клієнтів.

Використання алгоритму Round Robin не вимагає зв'язку між серверами, тому він може використовуватися як для локального, так і для глобального балансування.

Рішення на базі алгоритму Round Robin відрізняються низькою вартістю: для запуску роботи, досить просто додати кілька записів в DNS.

Алгоритм Round Robin має цілий ряд істотних недоліків. Щоб розподіл навантаження за цим алгоритмом відповідав згаданим вище критеріями справедливості та ефективності, потрібно, щоб у кожного сервера був наявності однаковий набір ресурсів. При виконанні всіх операцій також має бути задіяно однакову кількість ресурсів. У реальній практиці ці умови в більшості випадків виявляються нездійсненними.

Також при балансуванні за алгоритмом Round Robin абсолютно не враховується завантаженість того чи іншого сервера в складі кластера. Уявімо собі наступну гіпотетичну ситуацію: один з вузлів завантажений на 100%, в той час як інші - всього на 10-15%. Алгоритм Round Robin можливості виникнення такої ситуації не враховує в принципі, тому перевантажений вузол все одно буде ; гримувати запити. Ні про яку справедливість, ефективність та передбачуваність в такому випадку не може бути й мови.

Weighted Round Robin - це вдосконалена версія алгоритму Round Robin. Суть удосконалень полягає в наступному: кожного серверу присвоюється ваговий - коефіцієнт відповідно до його продуктивності і потужності. Це допомагає розподіляти навантаження більш гнучко, сервери з великою вагою обробляють більше запитів. Однак усіх проблем з відмовостійкістю це аж ніяк не вирішує. Більш ефективно балансування забезпечують інші методи, в яких при плануванні та розподілі навантаження враховується більшу кількість параметрів.



Вище ми перерахували основні недоліки алгоритму Round Robin. Назвемо ще один: у ньому абсолютно не враховується кількість активних на даний момент підключень.

Розглянемо практичний приклад. Є два сервера - позначимо їх умовно як А і Б. До сервера А підключено менше користувачів, ніж до сервера Б. При цьому сервер А виявляється більш перевантаженим. Підключення до сервера А підтримуються протягом більш довгого часу в порівнянні з підключеннями до сервера Б.

Дану проблему можна вирішити за допомогою алгоритму, відомого під назвою *least connections*. Він враховує кількість підключень, підтримуваних серверами в поточний момент часу. Кожен наступний запит передається серверу з найменшою кількістю активних підключень.

Існує вдосконалений варіант цього алгоритму, призначений в першу чергу для використання в кластерах, що складаються з серверів з різними технічними характеристиками і різною продуктивністю. Він називається *Weighted Least Connections*, який враховує при розподілі навантаження не тільки кількість активних підключень, а ваговий коефіцієнт серверів.

У числі інших удосконалених варіантів алгоритму *Least Connections* слід передусім виділити *Locality-Based Least Connection Scheduling* і *Locality-Based Least Connection Scheduling with Replication Scheduling*.

Перший метод був створений спеціально для кешуючих проксі-серверів. Суть полягає в наступному: найбільша кількість запитів передається серверам з найменшою кількістю активних підключень. За кожним з серверів податків закріплюється група клієнтських IP. Запити з цих IP направляються на «рідний» сервер, якщо він не завантажений повністю. В іншому випадку запит буде перенаправлений на інший сервер.

В алгоритмі *Locality-Based Least Connection Scheduling with Replication Scheduling* кожна IP-адреса або група IP-адрес закріплюється не за окремим сервером, а за цілою групою серверів. Запит передається найменш завантаженому з групи. Якщо всі сервери з «рідної» групи перевантажені, то буде : зарезервованій новий сервер. Цей сервер буде додано до групи, яка

обслуговує IP. з якого був відправлений запит. У свою чергу найбільш завантажений сервер з цієї групи буде видалено - це дозволяє уникнути надмірної реплікації.

Алгоритм Destination Hash Scheduling створений для роботи з кластером кешуючих проксі-серверів, але він часто використовується і в інших випадках. У цьому алгоритмі сервер, що обробляє запит, вибирається з статичної таблиці за IP-адресою одержувача.

Алгоритм Source Hash Scheduling ґрунтується на тих же самих принципах, що і попередній, тільки сервер, який буде обробляти запит, вибирається з таблиці за IP-адресою відправника.

Sticky Sessions - алгоритм розподілу вхідних запитів, при якому з'єднання передаються на один і той же сервер групи. Він використовується, наприклад, в веб-сервері Nginx. Сесії користувача можуть бути закріплені за конкретним сервером за допомогою методу IP hash. За допомогою цього методу запити розподіляються по серверах на основі IP-адреси клієнта. Як зазначено в документації, «метод гарантує, що запити одного і того ж клієнта буде передаватися на один і той же сервер». Якщо закріплений за конкретною адресою сервер недоступний, запит буде перенаправлений на інший сервер.

Застосування цього методу пов'язане з деякими проблемами. Проблеми з прив'язкою сесій можуть виникнути, якщо клієнт використовує динамічний IP. У прикладі, коли велика кількість запитів проходить через один проксі-сервер, балансування навряд чи можна назвати ефективним і справедливим. Описані проблеми, можна вирішити, використовуючи cookies. У комерційній версії Nginx є спеціальний модуль sticky, який використовує cookies для балансування. Тут є безкоштовні аналоги - наприклад, nginx-sticky-module.

## **Висновок до розділу 2**

Важливою частиною хмарної інфраструктури є балансування - навантаження. Виділено наступні класи рішень, використовуваних для балансування:

- балансування з пропуском трафіку через один пристрій балансування;
- балансування засобами кластера;
- балансування без пропуску трафіку через один пристрій балансування.

Балансування навантаження, або вирівнювання навантаження - метод розподілу завдань між декількома мережевими пристроями (наприклад, серверами) з метою оптимізації використання ресурсів, скорочення часу обслуговування запитів, горизонтального масштабування кластера (динамічне додавання/видалення пристроїв), а також забезпечення відмовостійкості і резервування). Балансування навантаження передбачає рівномірне навантаження обчислювальних вузлів.

У більшості випадків практичне і повне рішення задачі балансування завантаження складається з чотирьох кроків:

- Оцінка завантаження обчислювальних вузлів.
- Прийняття рішень про балансування.
- Ініціація балансування завантаження.
- Перенаправлення об'єктів.

Для досягнення рівномірного розподілу обчислювального навантаження необхідно враховувати насамперед поточне завантаження центрального процесора і ряд характеристик (обсяг доступної оперативної пам'яті, продуктивність шин даних та мережевого інтерфейсу, обсяг доступного дискового простору), вони можуть використовуватися як індикатори, що вказують на вихід апаратної платформи з допустимого режиму оптимального функціонування.

Балансування на мережевому рівні може здійснюватися за допомогою різноманітних способів:

- DNS-балансування;
- Побудова NLB-кластеру;

- Балансування за IP засобами вхідного шлюзу;
- Балансування засобами маршрутизаторів глобальної мережі.

Переваги балансування на третьому рівні:

- незалежність від протоколу високого рівня;
- абсолютна прозорість для серверів;
- незалежність від мережевого розташування серверів.

Недоліками балансування на третьому рівні відповідають способу реалізації балансування з єдиним пристроєм балансування або без єдиного пристрою балансування.

Для розподілу навантаження використовуються наступні алгоритми:

- Round Robin;
- Weighted Round Robin;
- Least Connections;
- Weighted Least Connections;
- Locality-Based Least Connection Scheduling;
- Locality-Based Least Connection Scheduling with Replication Scheduling;
- Destination Hash Scheduling;
- Source Hash Scheduling;
- Sticky Sessions.

Також, в фінансовому плані, програмний розподіл навантаження є дешевшим за апаратний. Оскільки, в його основу входить програмний продукт, він базується на описі програмного розподільвача навантаження.

Потрібно зауважити, що недоліком цього методу є те, що на кожен з серверів в кластерній системі потрібно встановлювати та налаштовувати цей комплекс, а також встановлювати додаткові мережеві карти, які будуть вимагати встановлення додаткового комутатора і велику кількість кабелів. Конфігурація кожної конкретної машини буде відрізнятися, адже кожен сервер матиме унікальне ім'я та ваговий коефіцієнт.

З позитивної точки зору, є те, що програмним розподілом навантаження можна керувати з будь-якого комп'ютера в мережі, не виникатиме проблем з додаванням нового сервера в кластер.

### 3. Конструкторська частина

#### 3.1 Підписка надання ресурсів «хмарним» сервісом

В дипломній роботі використовувалась безкоштовна студентська «хмарна» підписка Microsoft Azure в рамках програми DreamSpark(див. рис. 3.1). Веб- додатки служби Azure входять в повністю керовану «хмарну» платформу, яка дозволяє створювати і розгортати веб-додатки. Можливе використання ASP.NET, Java, PHP, Node.js або Python. Налаштування безперервну інтеграцію і розгортання за допомогою VSO, GitHub, TeamCity, Hudson або BitBucket. Це дозволяє автоматично будувати, тестувати і розгортати веб-додатки після кожного успішного тесту збірки коду або інтеграції.

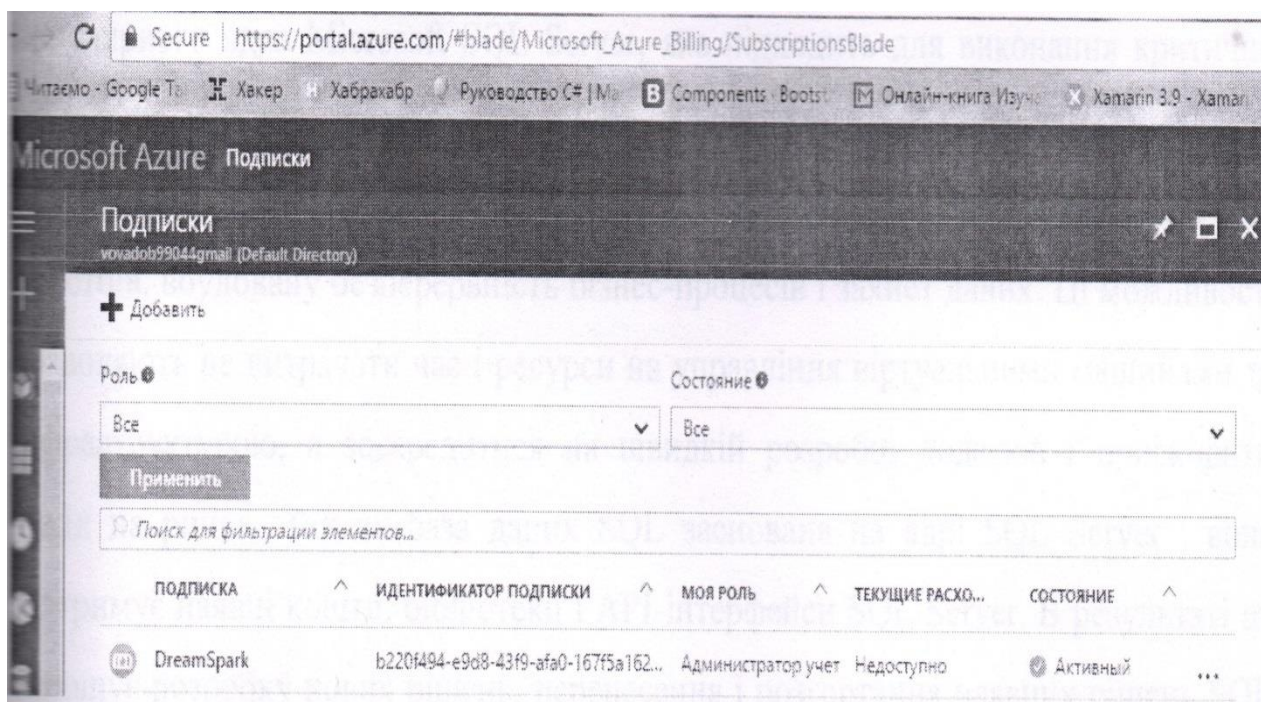


Рисунок 3.1 – «Хмарна» підписка Microsoft Azure

З СУБД MySQL від ClearDB веб-додатки отримують створення MySQL. За допомогою MySQL-рішення ClearDB можна розширити підтримку різних видів веб-додатків: CMS-рішень, Wordpress, Joomla, Acquia Drupal, phpBB та багатьох інших.

Application Insights надає повний огляд всіх параметрів доступності, продуктивності та використання служб ASP.NET, мобільних додатків для платформ WindowsPhone, iOS і Android. Функції пошуку та аналізу даних забезпечують постійне вдосконалення додатків.

Visual Studio Online являє собою найшвидший і простий спосіб планування, створення, постачання програмного забезпечення для різних платформ. «Хмарна» інфраструктура дозволяє зробити всі необхідні приготування для запуску, при цьому не доведеться займатися самостійно установкою і налаштуванням окремих серверів.

### **3.2 Проектування бази даних на «хмарі»**

База даних SQL - це реляційна служба баз даних в Microsoft Cloud на основі популярного ядра Microsoft SQL Server, що підходить для виконання критично важливих робочих навантажень. База даних SQL забезпечує прогнозовану продуктивність на декількох рівнях служби, динамічну масштабованість без простоїв, вбудовану безперервність бізнес-процесів і захист даних. Ці можливості дозволяють не витрачати час і ресурси на управління віртуальними машинами та інфраструктурою, а зосередитися на швидкій розробці додатків і прискорити Вихід на ринок. Так як база даних SQL заснована на ядрі SQL Server, вона підтримує наявні кошти, бібліотеки і API-інтерфейси SQL Server. В результаті це спрощує розробку нових рішень, перенесення і розгортання наявних рішень SQL Server в Microsoft Cloud, і для цього вам не потрібні нові навички.

Для проектування нашої бази даних був створити новий сервер, з використанням будь якого регіону(див. рис. 3.2.1).

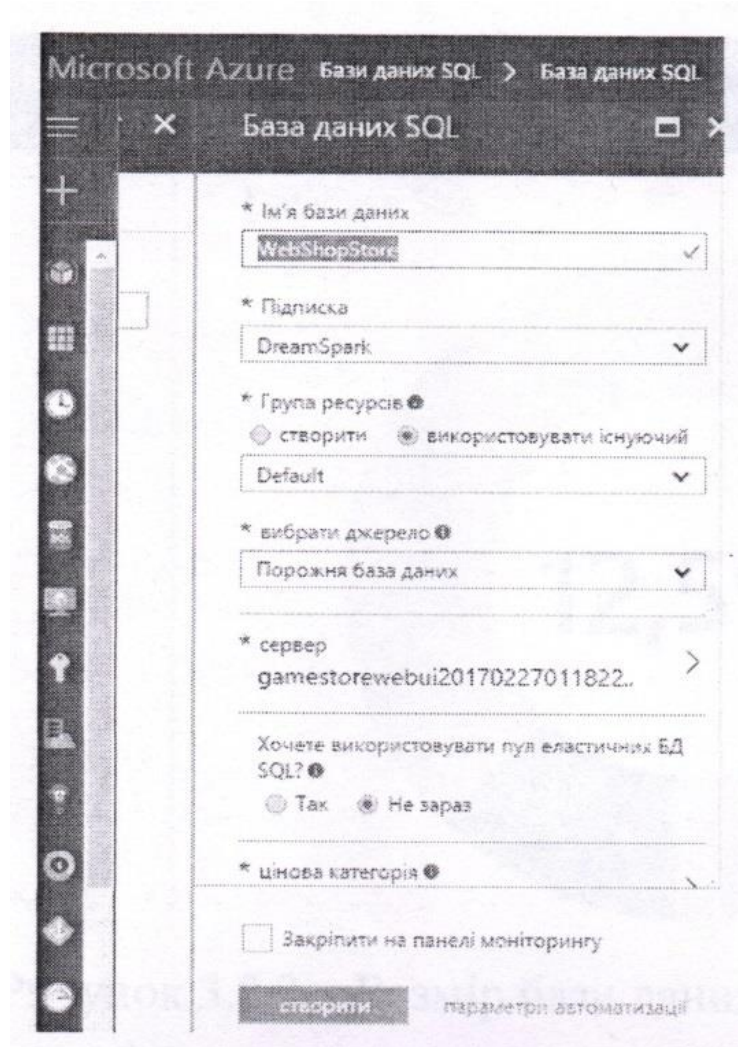


Рисунок 3.2.1 - Створено нову базу даних WebShopStore

Якщо жоден шаблон не задовольняє ваших потреб або потрібно використати дані з іншої програми, може бути краще створити базу даних із нуля.

Для створення нової бази необхідно вказати підписку, групу ресурсів(в такому можливо вибрати вже існуючу), шаблон бази даних, яка за замовчуванням є порожньою і обов'язково вказати цінову категорію.

WebShopStore містить такі налаштування:

- моніторинг;
- підтримка і усунення недоліків;
- діагностика і вирішення проблем;



- цінова категорія;
- журнал дій;
- розмір бази даних.

Розмірність бази даних є не значною 32 МБ, тому що підписка на дану «хмарну» систему є безплатною(див. рис. 3.2.2).

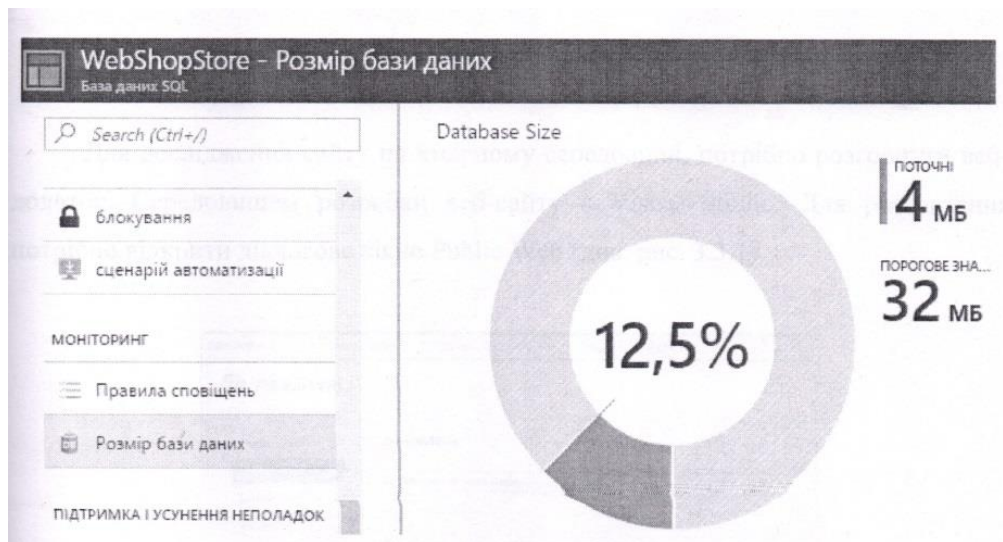


Рисунок 3.2.2 - Розмір бази даних

В результаті було проєктовано базу даних на «хмарному» сервісі, яка працює в групі ресурсів Azure і підключена до логічного сервера(див. рис. 3.2.3).

Атрибути розмірності мають бути:

- Мовними (етикетки, що складаються з повних слів)
- Описовими
- Закінченими (без відсутніх значень)
- Дискретно оцінюваними (має лише одне значення таблиці в рядку розмірностей)
- Забезпечені якістю (без правопису або неможливих значень)

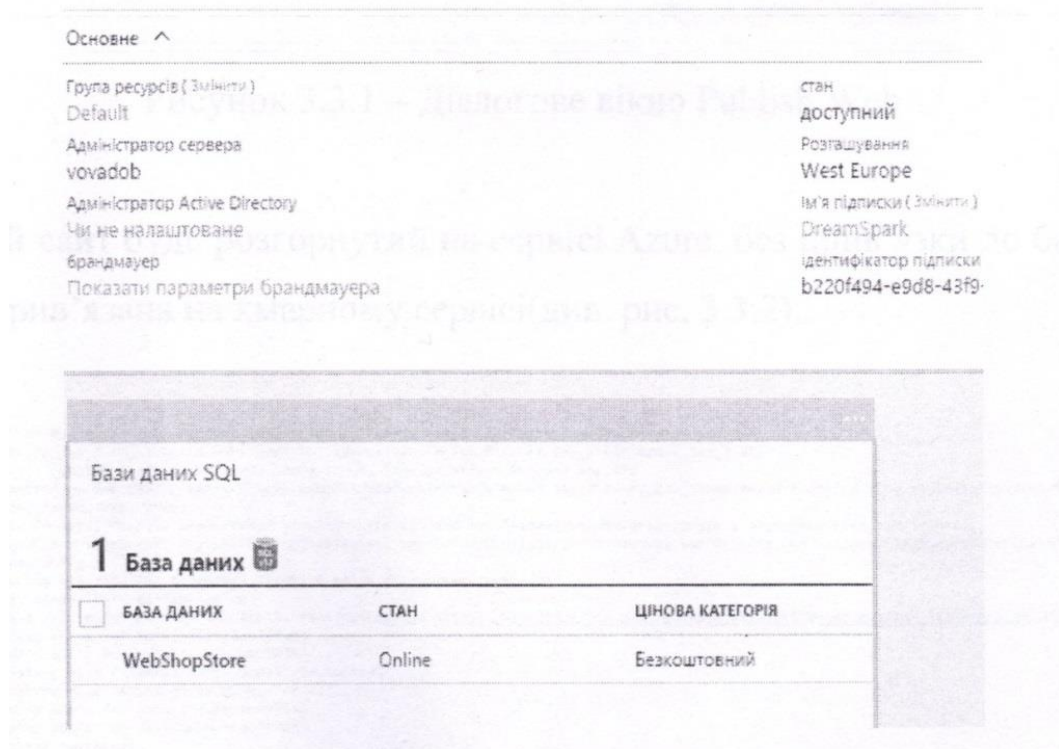


Рисунок 3.2.3 - Результат проектування бази даних

Також, було налаштовано брандмауер на рівні сервера, які дозволять суб'єкту даного сервера входити на інший сервер з двох зазначених IP-адрес.

### 3.3 Дослідження та розгортання веб-сайту в «хмарному» середовищі

Для дослідження сайту на «хмарному» середовищі, потрібно розгорнути веб- додаток. Середовищем розробки веб-сайту є Visual Studio.

Також, можна зазначити певні вимоги до клієнт-серверних додатків. Необхідно виділити декілька основних існуючих підходи при розгляді платформ:

- формування відповідей та обробка запитів;
- Вбудовування програмного коду в шаблони HTML сторінок.

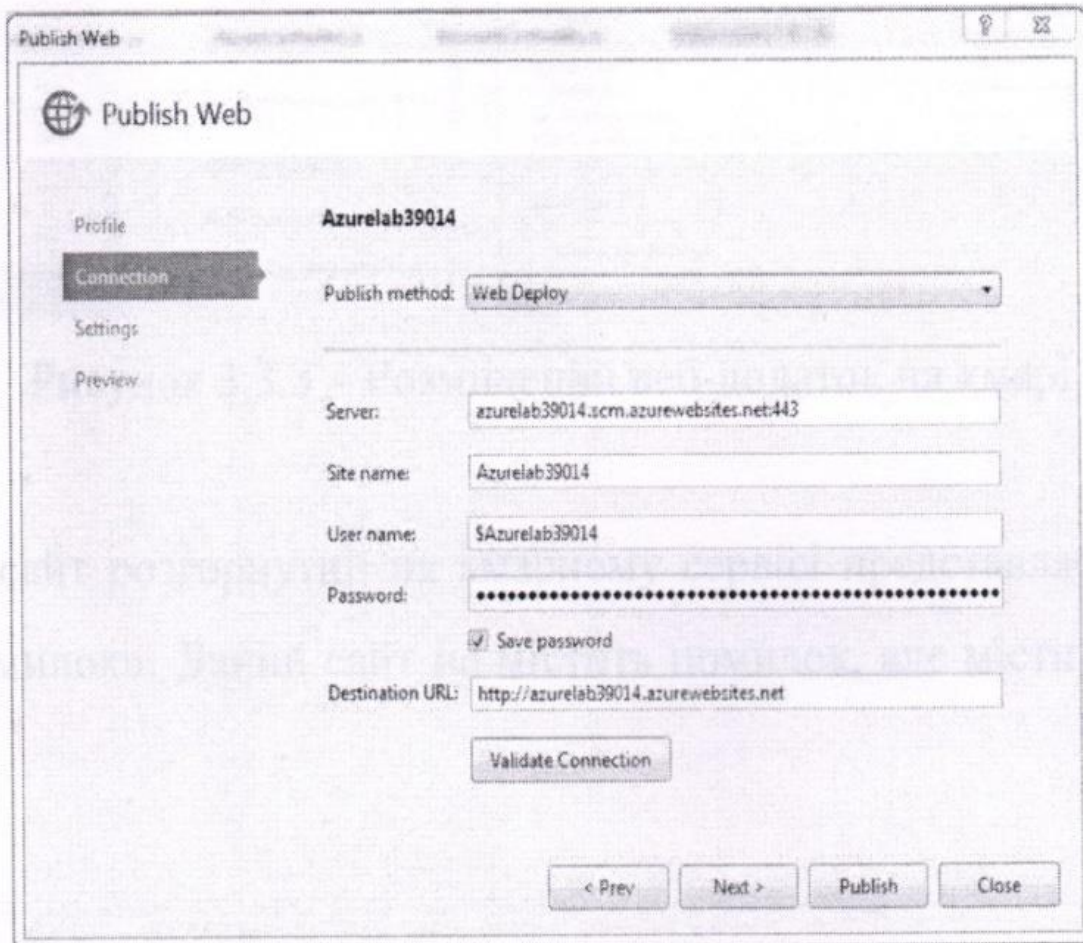


Рисунок 3.3.1 - Діалогове вікно Publish Web

Даний сайт буде розгорнутий на сервісі Azure, без прив'язки до бази даних. Вона буде прив'язана на «хмарному» сервісі(див. рис. 3.3.2)

```
1>----- Build started: Project: Azure_lab_3, Configuration: Release Any CPU -----
1> Azure_lab_3 -> C:\Users\Voiva\Desktop\TestWork\Azure_lab_3\Azure_lab_3\bin\Azure_lab_3.dll
2>----- Publish started: Project: Azure_lab_3, Configuration: Release Any CPU -----
2>Transformed web.config using C:\Users\Voiva\Desktop\TestWork\Azure_lab_3\Azure_lab_3\Web.Release.config into obj\Release\Transfo
  \transformed\web.config.
2>Auto ConnectionString Transformed Views\Web.config into obj\Release\CSAutoParameterize\transformed\Views\Web.config.
2>Auto ConnectionString Transformed obj\Release\TransformWebConfig\transformed\web.config into obj\Release\CSAutoParameterize\tra
  \Web.config.
2>Copying all files to temporary location below for package/publish:
2>obj\Release\Package\PackageTmp.
2>Start Web Deploy Publish the Application/package to https://azurelab39014.scm.azurewebsites.net/wdeploy.exe?site=Azurelab39014
2>Adding ACL's for path (Azurelab39014)
2>Adding ACL's for path (Azurelab39014)
2>Updating file (Azurelab39014\Views\Web.config).
2>Updating file (Azurelab39014\Web.config).
2>Adding ACL's for path (Azurelab39014)
2>Adding ACL's for path (Azurelab39014)
2>Publish Succeeded.
2>Web App was published successfully http://azurelab39014.azurewebsites.net/
***** Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped *****
***** Publish: 1 succeeded, 0 failed, 0 skipped *****
```

Рисунок 3.3.2 - Компіляція і розгортання веб-додатка на «хмарі».

Після успішної компіляції сайт буде розгорнутий на даному сервісі. Даний веб-до даток містить групу ресурсів, стан, розташування, ім'я підписки, UML-адреса, цінову категорію, дані FTP чи розгортання(див. рис. 3.3.3).

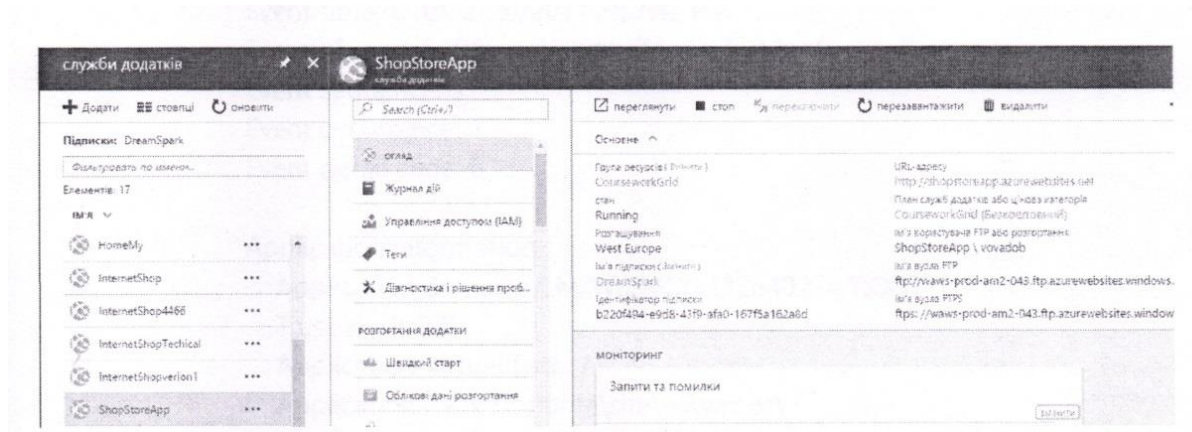


Рисунок 3.3.3 - Розміщений веб-додаток на «хмарі»

Кожний сайт розгорнутий на «хмарному» сервісі представляється діаграмою «Запитів та помилок». Цей сайт не містить помилок, але містить 2 запити(див. рис. 3.3.4).

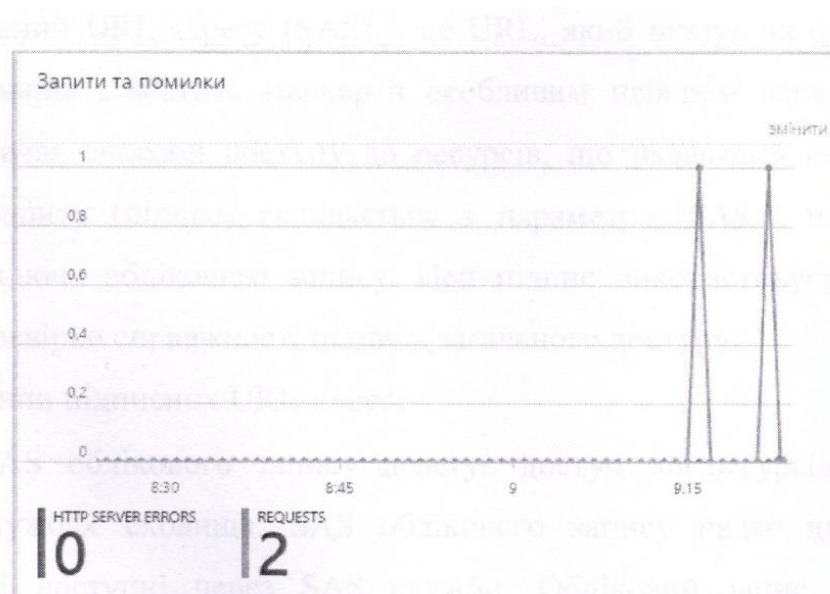


Рисунок 3.3.4 - Діаграма «Запитів та помилок»

Веб-додаток на «хмарі» містить Log для перевірки коректності роботи додатка, який містить в собі код події, повідомлення помилки, час, послідовність подій, виникнення нової події, детальний код події(див. рис. 3.3.5).

```
Event code: 3005
Event message: An unhandled exception has occurred,
Event time: 2/26/2017 2:57:42 PM
Event time (UTQ: 2/26/2017 2:57:42 PM
Event ID: 3 del 151 ff8bad43988bc71 a54e5912c45
Event sequence: 3
Event occurrence: 1
Event detail code: 0
Application information:

Application domain: /LM/W35VC/1332640274/ROOT-1
Trust level: Full Application Virtual Path: /
Application Path: D:\home\srtc\wwwroot\
Machine name: RD000D3A22C1QF
Process information:
Process ID: 6004
Process name: w3wp.bce
```

Рисунок 3.3.5 - Event Log веб-додатка

Підписаний URL-адресу (SAS) - це URL, що вказує на один або кілька ресурсів сховища і містить маркер з особливим набором параметрів запиту. Даний маркер визначає способи доступу до ресурсів, що надаються клієнту. Один з параметрів запиту (підпис) складається з параметрів SAS і підписується за допомогою ключа облікового запису. Даний підпис використовується сховищем Azure для перевірки справжності підпису загального доступу.

Є два типи підписних URL-адрес:

- SAS облікового запису делегує доступ до ресурсів в одній або декількох службах сховища. SAS облікового запису надає доступ до всіх операцій, які доступні через SAS служби. Обліковий запис SAS дозволяє

делегувати доступ до операцій, що належать до конкретної служби, таким як отримання і установка властивостей служби або отримання статистики служби. Крім того можна делегувати доступ до операцій читання, записи і видалення в контейнерах великих двійкових об'єктів, таблицях, чергах і загальних папках, які заборонені в SAS служби.

- SAS служби делегує доступ до ресурсу тільки в одній зі служб зберігання: службі BLOB-об'єктів, черг, таблиць або файлів.

Розглянемо розгорнутий веб-додаток за його URL-адресою (див. рис. 3.3.6).

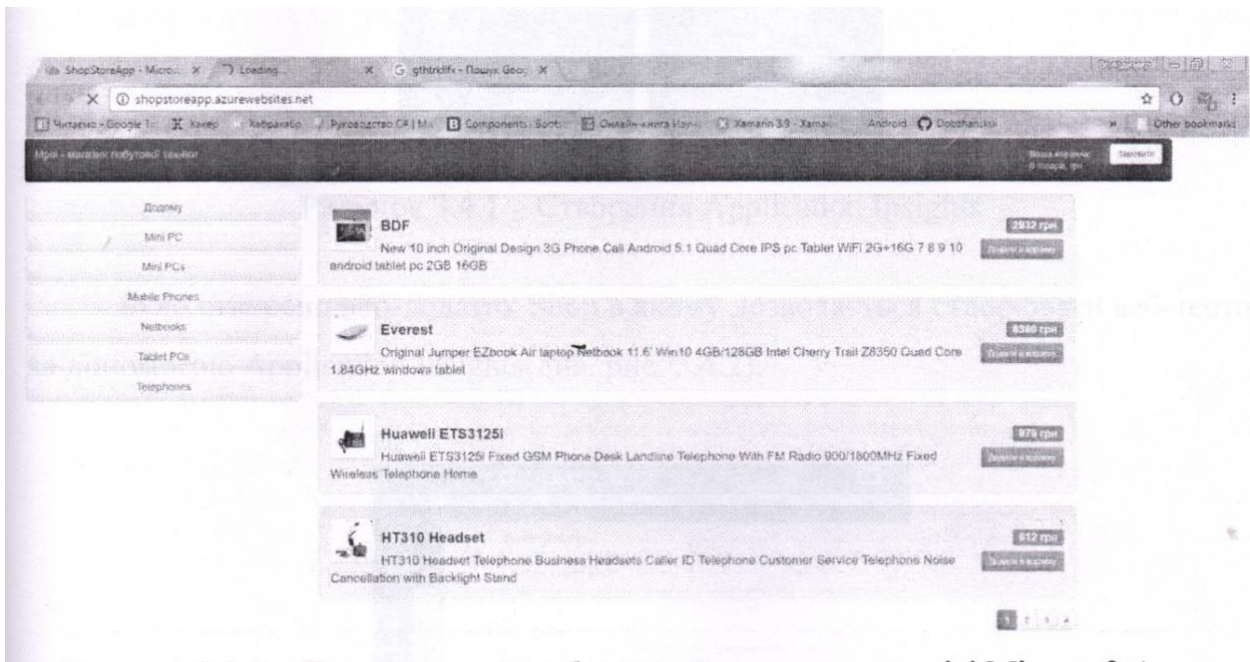


Рисунок 3.3.6 - Представлення веб-сайту на «хмарному» сервісі Microsoft Azure

### 3.4 Сервіси моніторингу для веб-додатків

Сервіси моніторингу це повний огляд всіх параметрів життєвого циклу програм і здійснення постійного контроль за працездатністю додатків. Одним з таких сервісів є Application Insights<sup>НВ</sup>. рис. 3.4.1).

На цей час, даний сервіс є частиною «хмарної» платформи Azure. Сервіс дозволяє моніторити показники використання, доступності та продуктивності додатка.

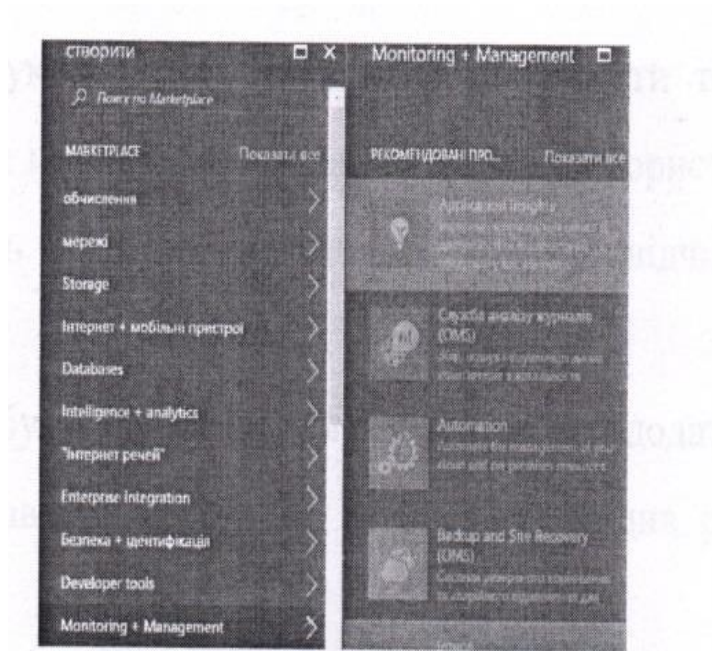


Рисунок 3.4.1 - Створення Application Insights

Створено веб-додаток Shop в якому дозволяється створювати веб-тести за допомогою Application Insights(див.рис. 3.4.2).

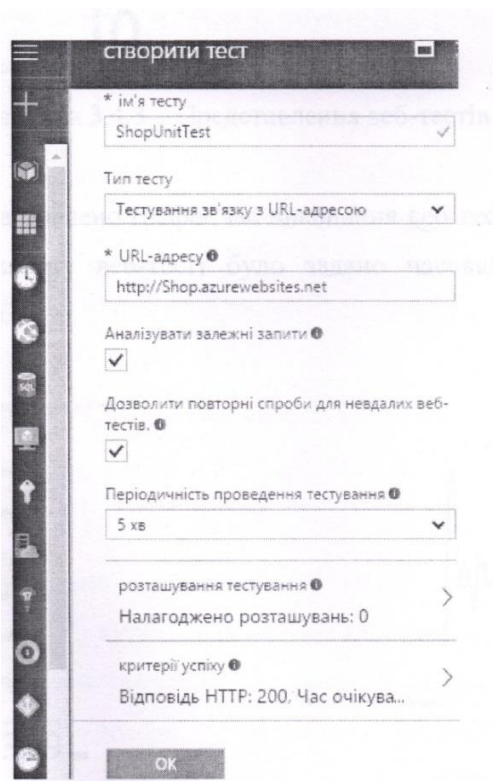


Рисунок 3.4.2 - Створення веб-тесту для Shop

Протестувати продуктивність програми можна на порталі Azure. Дані тести імітують призначену для користувача навантаження на додаток за певний період часу і оцінюють відгук додатка.

Наприклад, результати тесту показують, як швидко додаток реагує на звернення певного кількості користувачів. Вони також відображають кількість невдалих запитів, які можуть свідчити про проблеми з додатком.

Даний веб-тест був створений для перевірки веб-додатку на помилки. На даному графіку представлено час відгуку всіх веб-тестів(див. рис. 3.4.3).

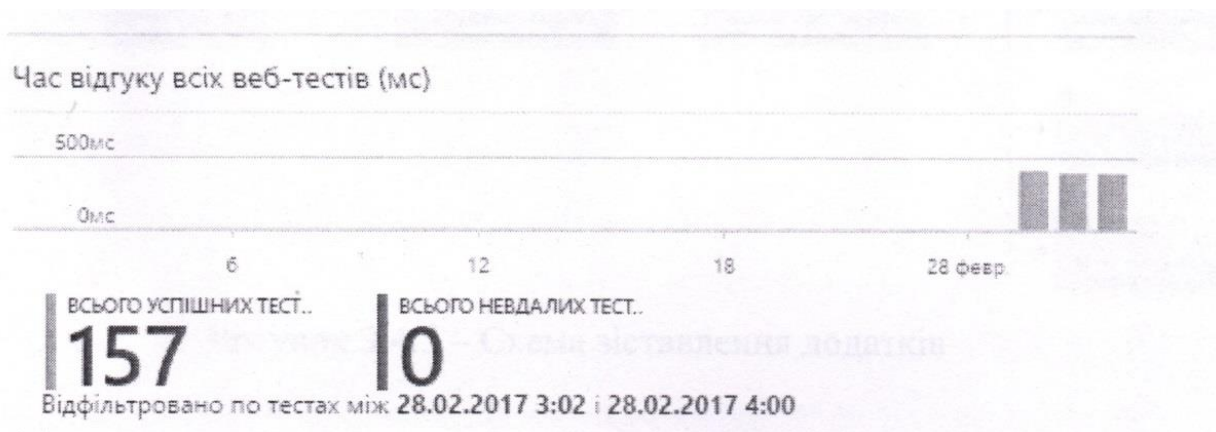


Рисунок 3.4.3 - Представлення веб-тестів

Крім того було представлено графік час виконання веб-тесту, який відображає відрізок часу. В даному веб-тесті було задано часовий діапазон в одну годину(див. рис. 3.4.4)

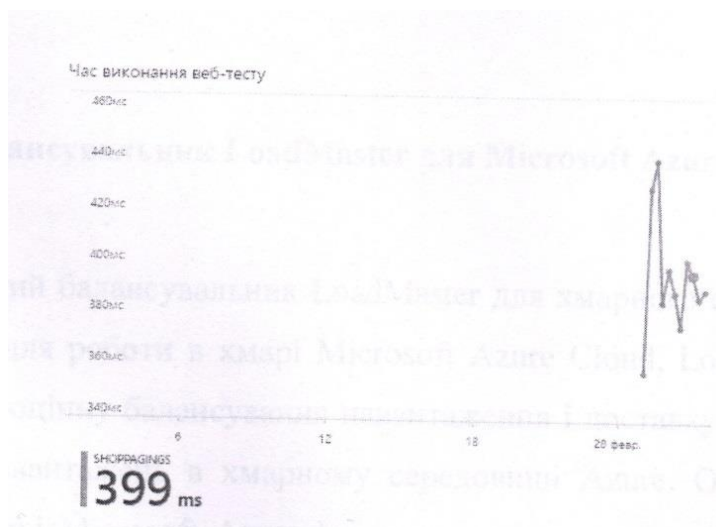


Рисунок 3.4.4 - Час виконання веб-сайту



Ще одним не менш важливим дослідженням є схема зіставлення додатків, яке визначає навантаження, продуктивність і частоту збоїв для певного компонента (див. рис. 3.4.5). Схема зіставлення додатків в Azure Application Insights - це наочний макет відносин залежності компонентів програми.

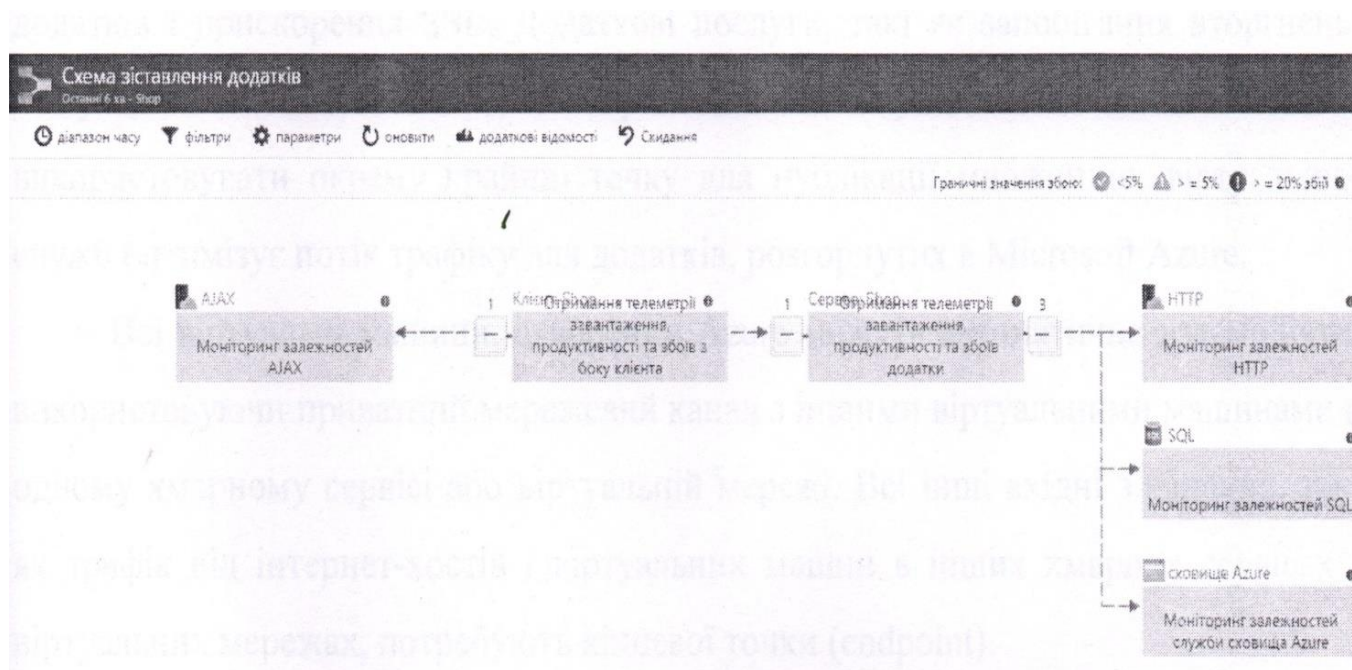


Рисунок 3.4.5 - Схема зіставлення додатків

За допомогою компонентів, вони відображають ключові показники ефективності, включаючи навантаження, продуктивність, збої і оповіщення, можливо визначати інші компоненти, що викликають проблеми з продуктивністю або збої. Коли додаток використовує служби Azure, можна також вибрати дані системи діагностики Azure, наприклад рекомендації Помічника по базах даних SQL.

### 3.5 Балансувальник LoadMaster для Microsoft Azure

Віртуальний балансувальник LoadMaster для «хмарного» середовища Azure - оптимізований для роботи в «хмарі» Microsoft Azure Cloud, LoadMaster для Azure забезпечує повноцінну балансування навантаження і доставку

додатків на рівнях для робочих навантажень в «хмарному» середовищі Azure. Об'єкти LoadMaster, розміщені в «хмарі» Microsoft Azure і в середовищах локальної приватної «хмари», працюють спільно для забезпечення безперервності доставки послуг на міжхмарному рівні [15].

Віртуальний балансувальник LoadMaster для Azure доповнює додатки, розміщені в інфраструктурі IaaS Microsoft Azure, надаючи повноцінний розподіл трафіку і безперебійність сеансу роботи на рівні L7, перевірку працездатності додатків і прискорення SSL. Додаткові послуги, такі як запобігання вторгнень, кешування і стиснення для опублікованих служб і здатність спільно використовувати окрему крайню точку для публікації множинних віртуальних служб оптимізує потік трафіку для додатків, розгорнутих в Microsoft Azure.

Всі віртуальні машини, створені в Azure можуть автоматично взаємодіяти, використовуючи приватний мережевий канал з іншими віртуальними машинами в одному «хмарному» сервісі або віртуальній мережі. Всі інші входні з'єднання, такі як трафік від інтернет-хостів і віртуальних машин в інших «хмарних» сервісах і віртуальних мережах, потребують кінцевої точки .

Кінцеві точки використовують для різних цілей. Використання за замовчуванням і конфігурації кінцевих точок на віртуальній машині, що ви створюєте з Azure Management Portal призначені для протоколу віддаленого робочого столу (RDP) і трафіку дистанційних сеансів Windows PowerShell. Дані кінцеві точки дозволяють дистанційно керувати віртуальною машиною через Інтернет.

Ще одне застосування кінцевих точок - настройка балансування навантаження Azure для поширення певного типу трафіку між декількома віртуальними машинами або сервісами. Наприклад, ви можете розподілити навантаження трафіку веб-запиту між декількома веб-серверами і веб-ролями.

Кожній кінцевій точці визначеній для віртуальної машини присвоюється публічний і приватний порт, або TCP або UDP. Інтернет-хости відправляють їх входний трафік на публічну IP адресу «хмарного» сервісу і до

публічного порту. Віртуальні машини і сервіси в рамках «хмарного» сервісу "слухають" приватну IP- адресу та приватний порт.

При налаштуванні балансування навантаження для трафіку між декількома віртуальними машинами або сервісами, Azure забезпечує випадковий розподіл вхідного трафіку. Для «хмарного» сервісу, який містить екземпляри веб-ролей або ролей працівників, можна визначити публічну кінцеву точку у визначенні сервісу. Для «хмарного» сервісу, який містить віртуальні машини, можна додати кінцеву точку на віртуальній машині при її створенні або додати кінцеву точку пізніше.

Наступний рисунок показує кінцеву точку балансування навантаження для зашифрованого веб-трафіку, який є спільним для трьох віртуальних машин для публічного та приватного порту TCP 443. Ці три віртуальні машини перебувають в збалансованості.

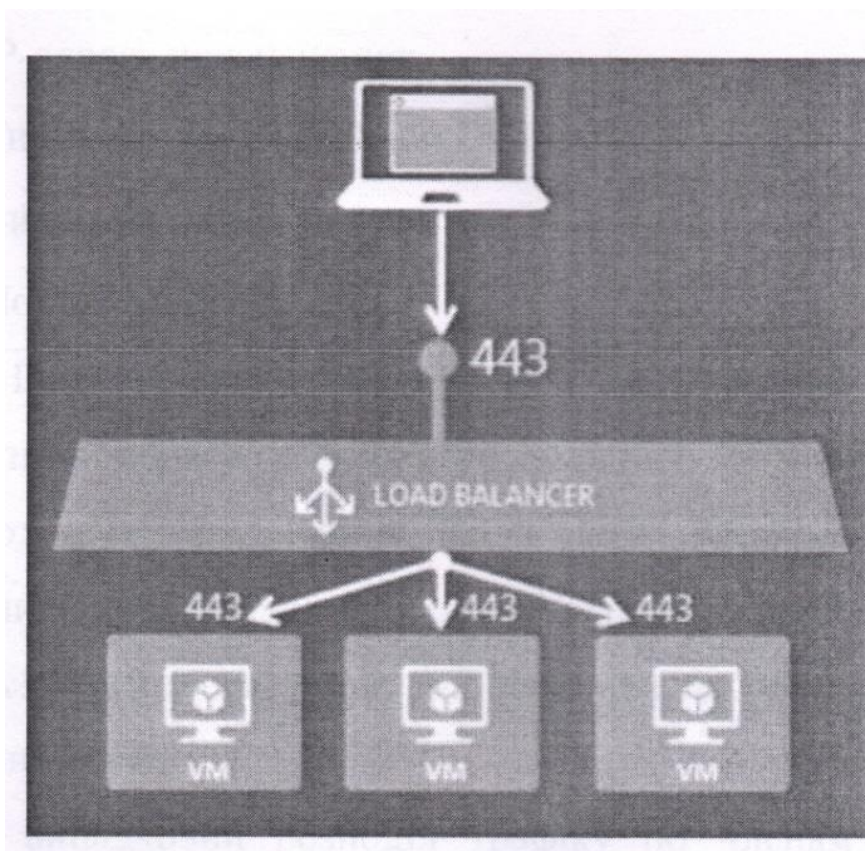


Рисунок 3.5.1 - Схема роботи Load Balancer

Коли Інтернет-клієнти відправляють запити до веб-сторінки на публічну IP адресу «хмарної» служби і TCP-порт 443, Azure Load Balancer виконує випадкове балансування цих запитів між трьома віртуальними машинами в наборі.

Azure також підтримує внутрішнє балансування навантаження трафіку між віртуальними машинами в межах «хмарного» сервісу, між віртуальними машинами в «хмарних» сервісах що містяться у віртуальній мережі, а також між локальними комп'ютерами і віртуальними машинами в віртуальній мережі.

Azure Load Balancer працює на рівні 4, транспортному рівні моделі OSI. Це означає, що вона працює на окремих потоках TCP або UDP трафіку, як це визначено в їх джерелі і IP-адресами призначення і номерами портів. Балансування навантаження на рівні 4 гарантує, що всі пакети для даного з'єднання TCP або UDP обміну повідомленнями направляються до однієї цілі.

Azure Load Balancer розподіляє навантаження серед декількох доступних серверів (віртуальних машин) шляхом обчислення хеш-функцій на трафіку, отриманого на заданій вхідній кінцевій точці. Azure Load Balancer використовує наступні поля з вхідного пакета для обчислення значення хеш:

- Вихідна IP-адреса;
- IP-адреса призначення;
- Тип протоколу (TCP або UDP);
- Вихідний порт;
- Порт призначення.

Azure Load Balancer потім використовує це значення для направлення трафіку на доступний сервер. Всі пакети з однієї TCP або UDP передачі, направляється на один сервер в наборі. Коли клієнт закриває і знову відкриває з'єднання або починає новий сеанс з того ж IP-адресу джерела, порт джерела, як правило, змінюється. Так створюється нове й інше значення хеш та новий маршрут трафіку на доступний сервер.

Результат - випадковий розподіл трафіку по членам зі збалансованої множини. На підставі цього випадкового розподілу, можливо для різних з'єднань отримати направлення на один і той же сервер.

### 3.5.1 Налаштування балансування навантаження в Azure Portal

Створюємо новий Load Balancer. В Create load balancer, введіть ім'я для балансування навантаження, тип навантаження можна вибрати public чи internal, створення зовнішня ip, вибір групи ресурсів, розташування даного балансувальника. Балансування навантаження буде розгортатися(див. рис. 3.5.2).

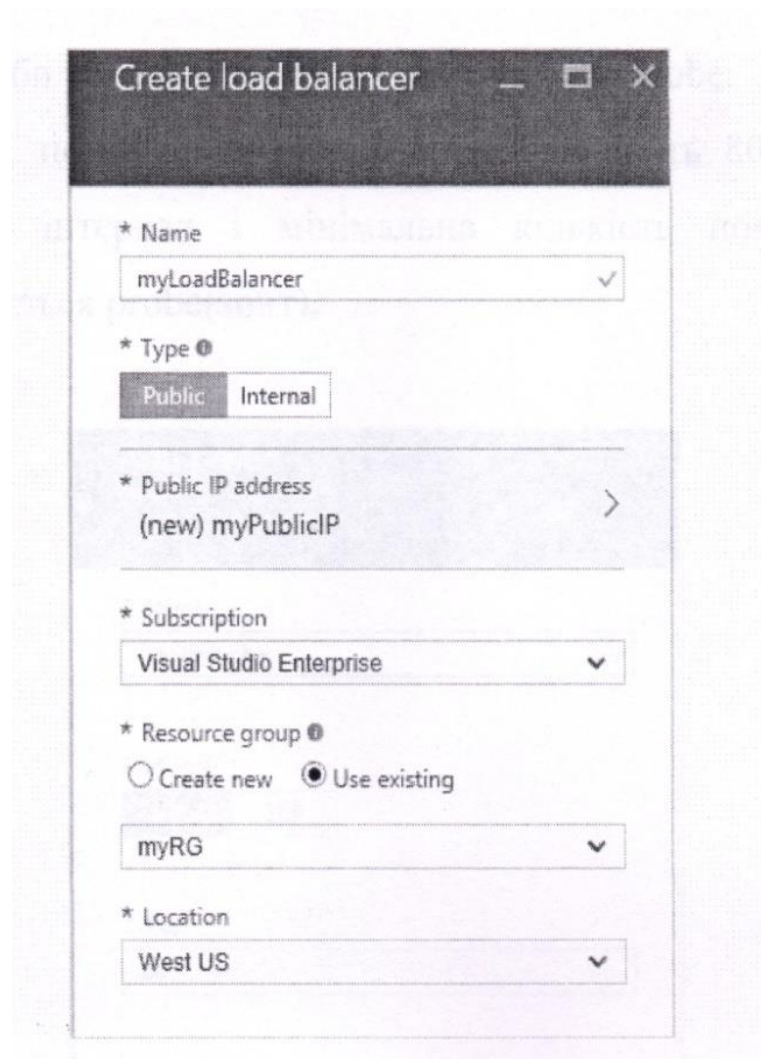


Рисунок 3.5.2 - Створення нового Load Balancer

Після успішного розгортання, заповнення поля імені back-end address pool. Додано backend pool. Наступним кроком є вибір віртуальних машин ( двох віртуальних машин ) , створених для балансування навантаження (див. рис. 3.5.3)

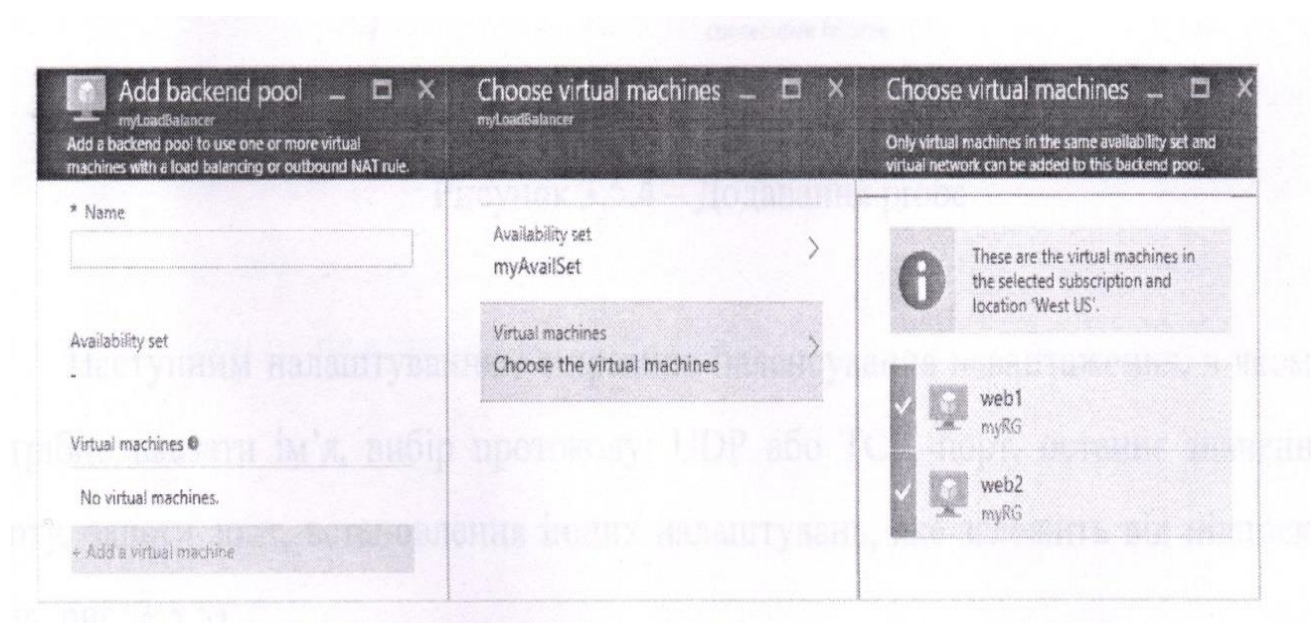


Рисунок 3.5.3 - Створення backend pool

Probe дозволяє отримувати детальний контроль про стан моніторингу існування для кінцевого пулу серверів. А для детального налаштування можна отримати помилкові запити і налаштувати час виходу.

Можна виділити два способи налаштування протоколу Probe: HTTP або TCP. Налаштування імені, порта за замовчуванням становить 80, задання шляху з розширенням .aspx, інтервал і мінімальна кількість помилок. Після усіх налаштувань створюється probe (зонт).

The image shows a software window titled "Add probe" from "myLoadBalancer". It contains the following configuration fields:

- Name:** A dropdown menu with "HealthProbe" selected.
- Protocol:** Two radio buttons, "HTTP" (selected) and "TCP".
- Port:** A text input field containing "80".
- Path:** A dropdown menu with "HealthProbe.aspx" selected.
- Interval:** A dropdown menu with "15" selected, with the unit "seconds" indicated below.
- Unhealthy threshold:** A text input field containing "2", with the unit "consecutive failures" indicated below.

Рисунок 3.5.4 - Додавання probe

Наступним налаштуванням є правило балансування навантаження, в якому потрібно вказати ім'я, вибір протоколу: UDP або TCP, порт, останнє значення порту, задати зонт, встановлення інших налаштувань, яке залежить від підписки (див. рис. 3.5.5).

The screenshot shows a configuration window for adding a load balancing rule. The fields are as follows:

- Name:** HTTP
- Protocol:** TCP
- Port:** 80
- Backend port:** 80
- Backend pool:** LB-backend (2 virtual machines)
- Probe:** HealthProbe (HTTP:80/HealthProbe.aspx)
- Session persistence:** None
- Idle timeout (minutes):** 4
- Floating IP (direct server return):** Disabled

Рисунок 3.5.5 - Правило балансування навантаження

NAT - це механізм зміни мережної адреси в заголовках IP датаграм, поки вони проходять через маршрутизуючий пристрій з метою відображення одного адресного простору в інший.

Налаштування відхідного правила підключень до NAT, створюємо назву правила, вибір сервісу, визначення цілі в такому випадку буде віртуальна машина, вибір протоколу, встановлення порту, встановлення порту віртуальної машини і зберігаємо це правило(див. рис. 3.5.6)



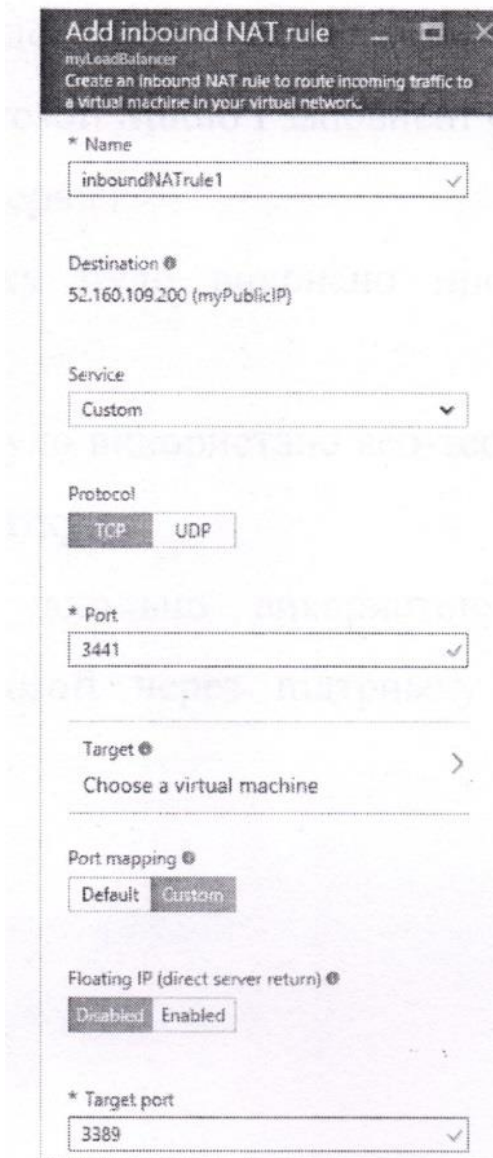


Рисунок 3.5.6 - Додавання вхідного правила підключення до NAT

Основна операція NAT полягає в наступному. Адреси всередині заглушки домену можуть бути повторно використані будь-яким іншим доменом. Наприклад, одна адреса класу А може бути використана багатьма доменами. На кожному виході точці між заглушкою і області хребта, встановлений NAT.

### Висновок до розділу 3

В даному розділі проаналізовано: підписку надання ресурсів від хмарного сервісу, описано дії проектування бази даних на «хмарі», розміщено і

проаналізовано веб-додаток на «хмарі», сервіси моніторингу для веб-додатків і представлено балансувальних LoadMaster від Microsoft Azure.

Підписка від DreamSpark є безкоштовно, тому більшість ресурсів не доступні для використання.

Базу даних на «хмарі» представлено у вигляді діаграми Database Size, таблиці додані за допомогою SQL Microsoft Studio і заповнені проміжними значеннями на випадок помилок в «хмарному» сервісі.

Розгортання веб-додатку виконано програмою Visual Studio, а проаналізовано на «хмарі».

В сервісі моніторингу було використано веб-тести, для перегляду помилок і швидкості запитів до веб-додатку.

Azure Load Balancer доцільно використовувати при комплексному балансуванні продуктів Microsoft через підтримку та взаємну вбудованість сервісів.

Проаналізовано такі засоби побудови систем прийняття рішень, простори даних, сховищ, також аналіз галузі комп'ютенгу.

База даних , сховища даних і простори саме це є науковою новизною статті .

Практичне значення полягає в зв'язку прийняття рішень та у визначенні задач компонент систем.

## 4. Науково дослідна частина

### 4.1 Мета, об'єкт, предмет та методи дослідження

На всіх етапах наукових досліджень використовуються інформаційно-комунікаційні технології. Формування навичок пошуку та аналізу наукової інформації, збору та обробки експериментальних даних засобами сучасних ІКТ та розвиток навичок роботи у співпраці в процесі наукового дослідження, саме це є основною метою даного курсу. Велику увагу заслуговують «хмарні технології». Їх головна перевага у науково-дослідній роботі є те, що вона надає динамічно масштабовані обчислювальні ресурси і програми через інтернет в якості сервісу під управлінням постачальника послуг.

В цей же час, відсутні методичні рекомендації з використання «хмарних» сервісів у наукових дослідженнях.

Суть «хмарних» обчислень полягає в наданні кінцевим користувачам віддаленого динамічного доступу до послуг, обчислювальних ресурсів і додатків через інтернет.

Мета нашого дослідження є розміщення розробленого веб-додатку на «хмарному» сервісі та аналіз веб-додатку на помилки.

Наукова діяльність аспірантів першого року навчання пов'язана переважно з пошуком, аналізом, узагальненням, систематизацією наукової літератури з проблеми обраного дослідження. Під час опитування та співбесіди виявили високий рівень інформатичної підготовки аспірантів та певний досвід використання «хмарних» сервісів. Більше 90% опитаних відзначали досвід з використання «хмарних» сховищ даних, таких як Google Диск, Drop Box, Sky Drive та інші. Усі без винятку опитані мали досвід роботи з пошуковими сервісами та системами пошуку наукових публікацій, такими як Google Академія, World Cat, SCIRUS тощо.

Для досягнення мети необхідно було вирішувати наступні наукові та практичні завдання:

- Аналіз відписки надання ресурсів «хмарним» сервісом ;
- Проектування баз даних на «хмарі»;

- Дослідження та розгортання веб-сайту на «хмарному» середовищі.

#### **4.2 Результати дослідження**

"Хмара" відкриває підхід до обчислень, при якому ні обладнання, ні програмне забезпечення не належать підприємству. Замість цього провайдер надає замовнику вже готовий сервіс.

Молоді компанії-стартапи, часто вдаються до допомоги «хмар» які потребують великих обчислювальних ресурсів для обслуговування користувачів, але не можуть дозволити собі створення і експлуатацію власного дата-центру.

Електронна пошта з веб-інтерфейсом стала одним з широкодоступних «хмарних» інтернет-сервісів. У цьому випадку всі дані зберігаються на віддалених серверах, а користувач отримує доступ до своїх листів через браузер з різних комп'ютерів або достатньо потужного мобільного пристрою.

Національним інститутом стандартів і технологій США встановлені такі обов'язкові характеристики «хмарних» обчислень :

- Універсальний доступ по мережі, послуги доступні споживачам через мережу передачі даних незалежно від термінального пристрою;

- Самообслуговування на вимогу, споживач самостійно визначає і змінює обчислювальні потреби, такі як серверний час, швидкості доступу та обробки даних, обсяг збережених даних без взаємодії з представником постачальника послуг;

- Об'єднання ресурсів постачальник послуг об'єднує ресурси для обслуговування великої кількості споживачів в єдиний пул для динамічного перерозподілу потужностей між споживачами в умовах постійної зміни попиту на потужності; при цьому споживачі контролюють тільки основні параметри послуги але фактичний розподіл ресурсів, що надаються споживачеві, здійснює постачальник;

-Еластичність, послуги можуть бути надані, розширені, звужені в будь-який момент часу, без додаткових витрат на взаємодію з постачальником, як правило, в автоматичному режимі;

-Облік споживання, постачальник послуг автоматично обчислює спожиті ресурси на певному рівні абстракції, і на основі цих даних оцінює обсяг наданих споживачам послуг.

Зручність і універсальність доступу забезпечується широкою доступністю послуг і підтримкою різного класу термінальних пристроїв.

Отже, «хмарні обчислення», в інформатиці - це модель забезпечення щоденного і зручного мережевого доступу на вимогу до загального пулу конфігуруючих обчислювальних ресурсів, які можуть бути надані і звільнені з мінімальними експлуатаційними витратами або зверненнями до провайдера.

## 5. Спеціальна частина

### 5.1 Огляд розробленого веб-додатку UML діаграмами

Даний веб-сайт представлений діаграмою прецедентів і діаграмою класів.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграма прецедентів (див. рис. 4.1), представлена взаємодією користувача, покупця і адміністратора з сайтом.

Користувач може виконувати такі дії:

- перегляд каталога товарів;
- реєстрація;

Покупець може виконувати такі дії:

- перегляд каталога товарів
- перегляд замовлень;

Адміністратор може виконувати такі дії:

- змінити статус замовлення;
- змінити каталог товарів.

Кожна з цих дій містить в собі групу функцій, можливість виконувати дії над різноманітними даним, які представлені на сайті, опрацьовувати їх і безпосередньо зберігати дані у базі.

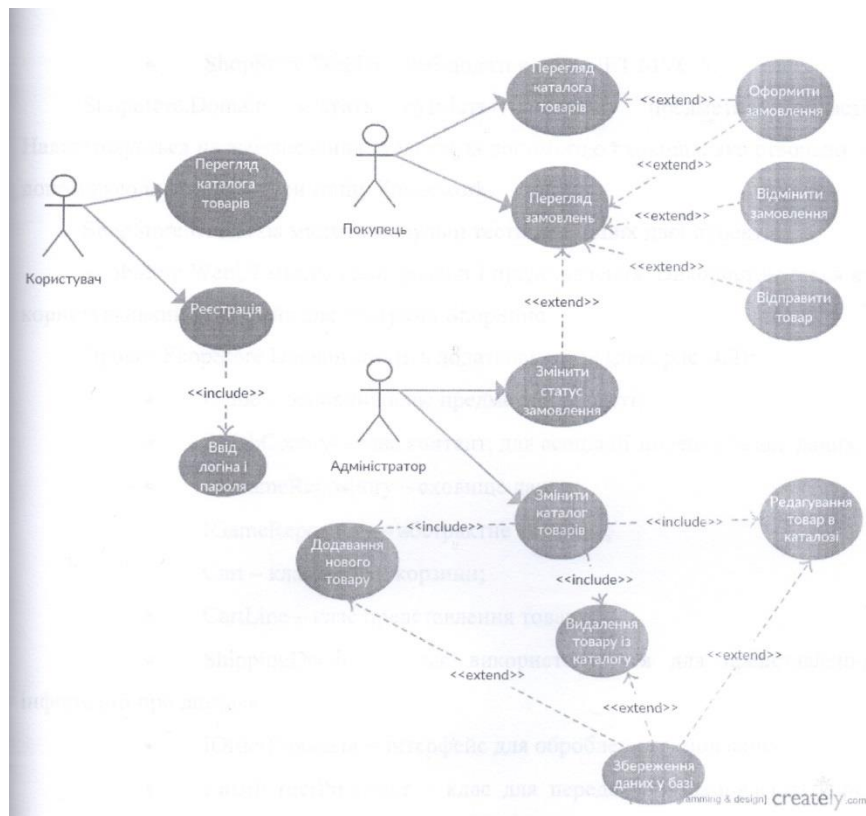


Рисунок 5.1- Діаграма прецедентів

Діаграма класів - це статичне представлення структури моделі. Вона відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, може містити позначення для пакетів та позначення для вкладених пакетів. Крім того, може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. На даній діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Даний веб-сайт складається з 3 проектів:

- ShopStore.Domain-бібліотека класів;
- ShopStore.UnitTests- проект модульного тестування;
- ShopStore.WebUI - веб-додаток ASP.NET MVC 5.

ShopStore.Domain містить сутність і логіку предметної області.

Налаштовується на забезпечення сталості за допомогою сховища, яке створено за допомогою інфраструктури Entity Framework.

ShopStore.UnitTests містить модульні тести для інших двох проектів.

ShopStore.WebUI містить контролери і представлення. Використовується як користувацький інтерфейс для програми ShopStore.

Проект ShopStore.Domain містить додаткові класи (див. рис. 4.2):

- Game - основний клас предметної області;

- EFGameRepository - сховище даних;

- IGameRepository - абстрактне сховище;

- EFDbContext - клас контент, для асоціації моделі з базою даних;

- Cart - клас моделі корзини;

- CartLine - клас представлення товару;

- ShippingDetails - клас використовується для представлення інформації про доставку;

- IOrderProcessor - інтерфейс для оброблення замовлення;

- EmailOrderProcessor - клас для передавання замовлення через електронну пошту;

- EmailSettings - клас містить всі налаштування для роботи з електронною поштою .



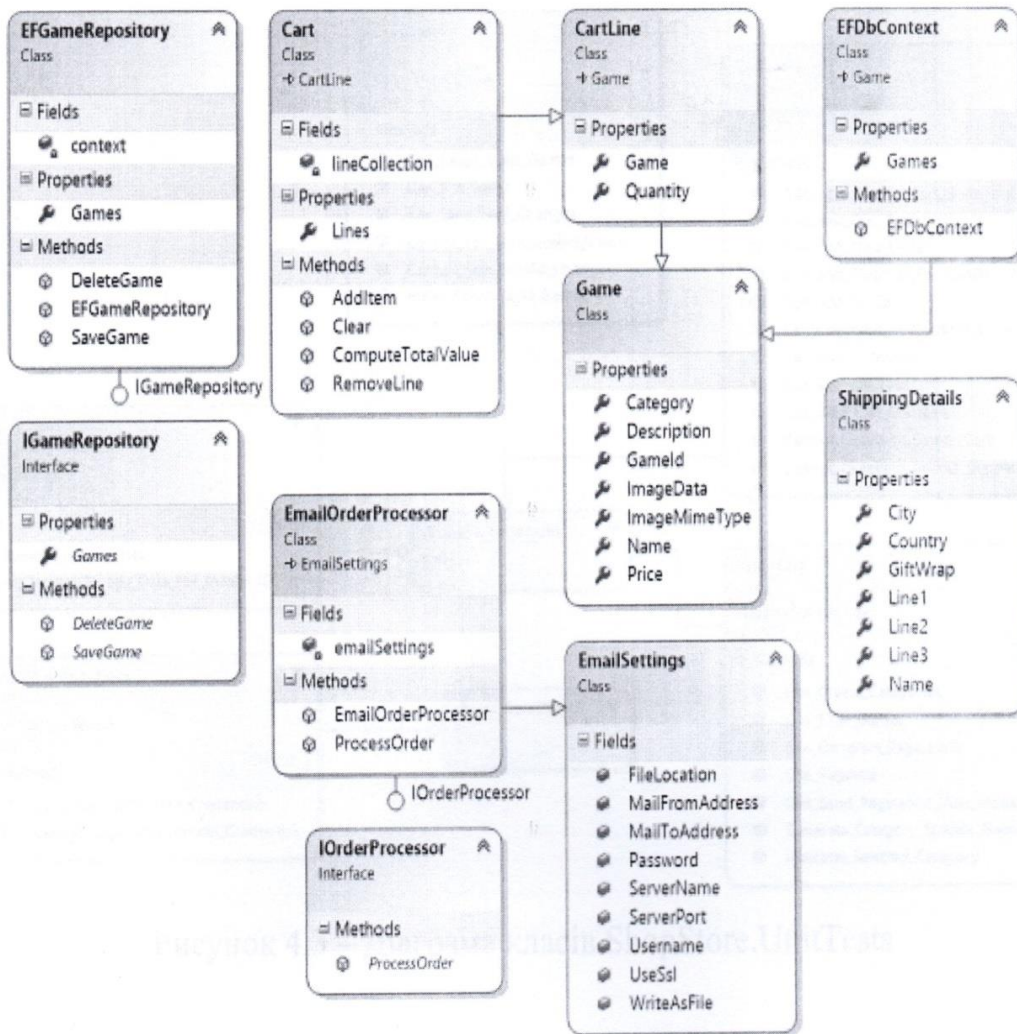


Рисунок 5.2 - Діаграма класів ShopStore.Domain

Клас Shop Store .UnitTests містить додаткові класи (див. рис. 4.3):

- MainTest - клас модульного тестування для перевірки даних на сторінках;

- AdminTests - перевірка коректного отримання даних Game, які знаходяться в сховищі, редагування та видаленню товару з адмін панелі;

- CartTests - перевірки: корзини, обробки замовлення;

- ImageTests - перевірка коректного типу MIME із сховища і перевірка на повернення не існуючого товару;

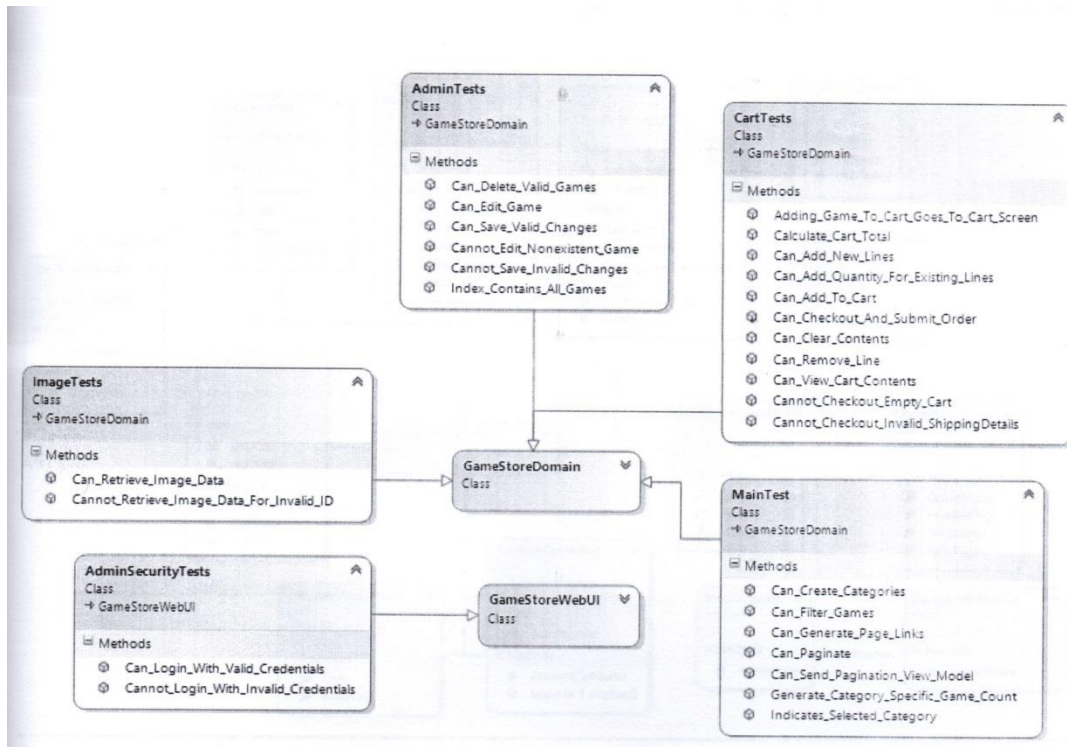


Рисунок 5.3 - Діаграма класів ShopStore.UnitTests

Клас ShopStore.WebUI містить додаткові класи (див. рис. 4.4):

- GameController - контроллер для отримання даних з класу Game;
- PagingInfo - клас для преставлення інформації про кількості доступних сторінок;
- PagingHelper - статичний клас для роботи з методом HTML;
- GamesListViewModel - клас представлення модулі екземпляра класу PagingInfo;
- NavController - контролер меню;
- CartController - контролер корзина;
- CartIndexViewModel - клас для відображення вмісту корзини;
- CartModelBinder - створення спеціального зв'язувана моделі;
- CartController - контролер корзина;
- CartIndexViewModel - клас для відображення вмісту корзини;

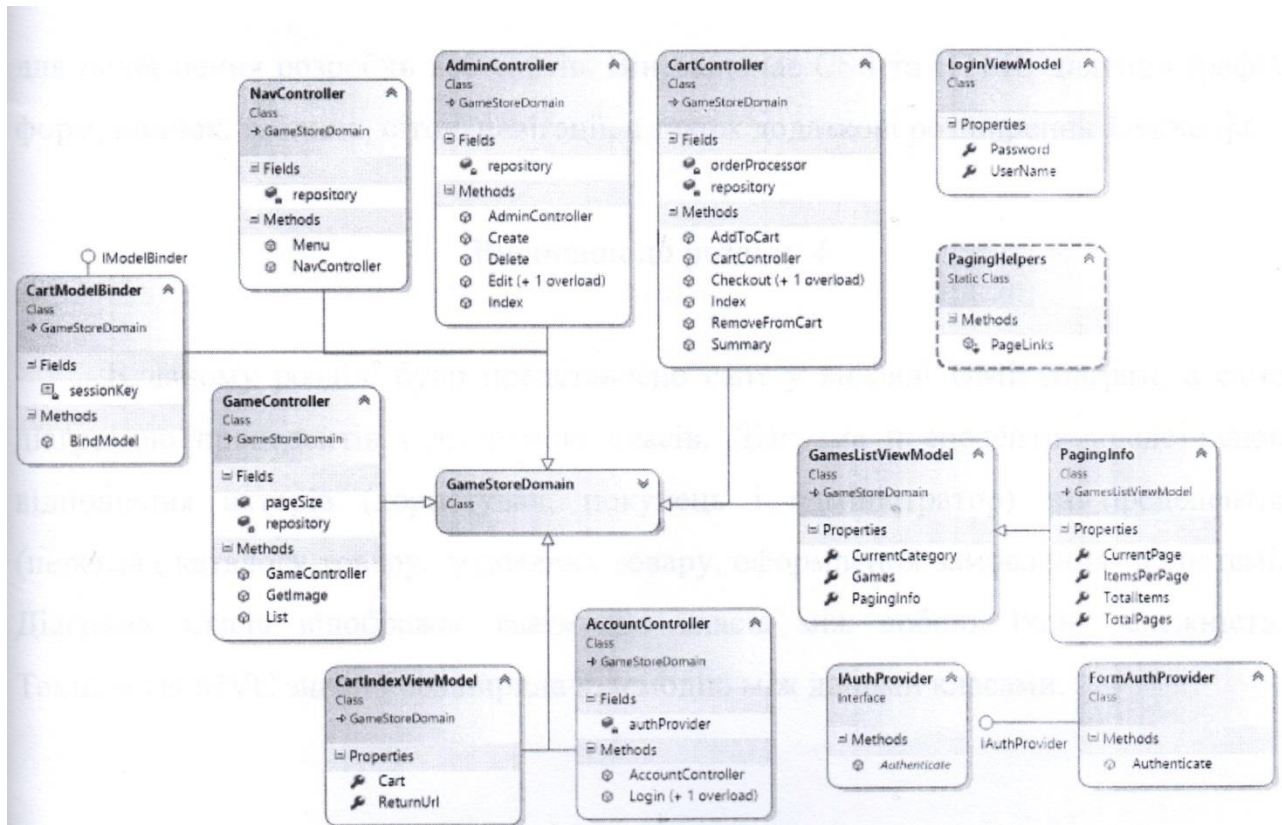


Рисунок 5.4 - Діаграма класів ShopStore.WebUI

-AdminController - контролер для підтримки роботи адміністратора по роботі з каталогами товарів;

-IAuthProvider - інтерфейс аутентифікації;

-LoginViewModel - клас моделі преставлення Account;

-Forms Authentication - клас аутентифікації;

-AccountController - контролер для обробки аутентифікації;

На сайті додатково використовуються такі технології: JQuery, CSS, HTML.

HTML - мова гіпертекстової розмітки документів у Інтернеті, призначення для написання веб-сайтів. Більшість веб-сторінок містять опис на мові розмітки HTML.

CSS використовуються для візуальної презентації сторінок. За допомогою зовнішніх таблиць стилів, внутрішніх таблиць стилів і стилів окремих елементів. Основним інструментом для розробки інтерфейсу був Bootstrap, він створений для полегшення розробки веб-сайтів.

Включає CSS та HTML для топографії, форм, кнопок, таблиць, сіток, навігації, а також додаткові розширення JavaScript.

### **Висновок до розділу 5**

У цьому розділі було представлено сайт у вигляді UML діаграм, а саме діаграмою прецедентів і діаграмою класів. Дана діаграма представляє відношення акторів (користувач, покупець і адміністратор) та прецедентів (перегляд каталогу товару, додавання товару, оформлення замовлення) в системі. Діаграма класів відображає взаємодію класів між собою, їхню залежність. Технологія MVC значно розширила взаємодію між даними класами.

## 6. Обґрунтування економічної ефективності

### 6.1 Планування товару

Планування товарів - це систематичне прийняття рішень з усіх аспектів розробки і управління продукцією підприємства, включаючи створення торгової марки, упаковки, систем збуту, сервісу. Представлено такі способи планування товарів:

-узагальнений - визначення того, яку користь матиме товар для споживача, його основної ідеї та мети використання;

-конкретний - визначення фізичної сутності товару, його характеристик (габарити, маса, надійність, колір, потужність);

-розширений - визначення всього, що становить образ продукції та порядок її обслуговування (технічні рекомендації, гарантії, умови повернення).

Згідно з цим розрізняють і три рівні створення товару:

-товар за задумом - констатується та вигода, яку отримає споживач від використання даного товару;

-товар у реальному виконанні - визначають рівень якості, комплекс властивостей та характеристик, специфічне оформлення, марочну назву;

-товар з підкріпленням - планують не тільки конкретний товар, але й умови його монтажу, поставки, торговельного кредитування, сервісу, гарантії, експлуатації.

На першій фазі визначається мета розробки нових товарів. Зазвичай, зони розробляються і запроваджуються відповідно до ринкових (забезпечення відповідної частки ринку, проникнення на нові ринки чи сегменти, забезпечення іміджу підприємства-новатора) чи внутрішніх потреб підприємства (забезпечення доходів та прибутку, завантаження виробничих потужностей чи власних каналів збуту).

Метою цього етапу є визначення ринків, для яких розробляється даний товар, та його стратегічної ролі для підприємства й споживачів.

Відповідно до визначеної мети на наступній фазі здійснюється розробка ідей. Для більшості підприємств у країнах з розвинутою економікою

генерування ідей нових товарів - це постійний пошук таких ідей, створення відповідних систем і методів їх вибору і нагромадження.

Відома досить велика кількість методів розробки ідей нових товарів. Найпоширеніші з них такі:

-опитування споживачів, посередників, працівників самого підприємства. Дане опитування проводять служби маркетингових досліджень підприємства. При цьому створюються спеціальні системи стимулювання подання відповідних пропозицій, активізації інтелектуальної діяльності респондентів;

-метод синектики – це використання аналогій з інших сфер життя, живої природи;

-«мозкова атака» («мозковий штурм») - метою якої є пошуку ідей нових товарів. Однією з основних особливостей мозкового штурму є те, що критика запропонованих ідей не допускається, щоб не стримувати творчої активності учасників. Крім цього, присутні в певний спосіб стимулюються для того, щоб вони висловлювали як найбільше оригінальних, навіть фантастичних ідей. Мозковий штурм буває класичний, анонімний, конструктивний, деструктивний, співпраці;

-стеження за діяльністю конкурентів. Даний метод дає можливість виявити факти успіху чи невдач конкурентів, проаналізувати їхні причини і в дальшому, вносячи відповідні корективи, творчо використати цю інформацію в діяльності власного підприємства;

-метод контрольних запитань передбачає складення переліку запитань стосовно того, наприклад, що можна поліпшити в певному конкретному товарі чи послугі. Такі запитання пропонуються для відповіді спеціальній групі експертів;

-морфологічний аналіз - це виділення найважливіших параметрів товару (наприклад, форма, матеріал і місткість упаковки) та вивчення всіх можливих співвідношень між ними;

-метод поєднань - порівнювання кількох товарів з метою знайти взаємозв'язок між ними і новим товаром.

Основна мета даної фази - створення достатнього запасу ідей для наступного відбору найдієвіших, найсвоєчасніших та найефективніших.

Ідеї нових товарів потребують ретельного відбору (фільтрації), до того ж оцінювання їхніх можливостей і перспектив реалізації. Для цього можуть бути використані критерії, пов'язані з:

- товарами - інтенсивність споживання, асортимент, рівень патентного захисту, можлива тривалість життєвого циклу, міра технічної складності та емоційного впливу на споживачів; якість, співвідношення «якість - ціна»;

- ринком - місткість, тенденції її збільшення, ринковий потенціал, кон'юнктура, сезонні коливання, стан конкуренції, поведінка конкурентів, сегменти ринку, наявний попит, поведінка споживачів, соціально-політичні ризики;

- виробництвом - стан ринку закупівель, обсяги капіталовкладень, терміни завершення основних етапів роботи, можливі труднощі фінансового та кадрового характеру, стан виробничої бази;

- просуванням та збутом - прибутковість, рентабельність, стан систем розподілу, стосунки зі сферою торгівлі, досвід власного торговельного апарату, витрати на просування.

На четвертій фазі планування товарів здійснюється розробка і перевірка їхнього задуму, тобто опрацьованого варіанта ідеї, вираженого зрозумілими і значущими для споживача поняттями. Часто для цього використовують письмові описи товару, рисунки, макети, ескізи тощо. При цьому головною ідеєю етапу є спроба визначити ставлення споживачів до концепції товару, можливе коло тих, хто ним користуватиметься і з якою метою.

Головними критеріями перевірки задуму товару є такі:

- комунікабельність - чи зрозуміють споживачі користь і вигоду від товару чи побачать її наочно;

- довіра - чи повірять споживачі в те, що цей товар сприятиме розв'язанню їхніх проблем;

- споживчий розрив - чи задовольняє потребу у виробництві саме такого товару решта постачальників продукції виробничо-технічного призначення;

-співвідношення ціни і вартості - чи відповідатиме, з погляду споживачів, ціна нового товару його справжній вартості;

-використання - як споживачі використовуватимуть цей товар.

Розробка є одним найважливіших фаз планування нових товарів стратегій маркетингу. Тут востаннє перевіряється доцільність вкладання коштів у створення моделі, зразка чи пробної партії продукції, а насамперед економічність виробництва і збуту товару.

Для визначення економічності виробництва і збуту нового товару оцінюють також комерційні ризики, величину необхідних інвестицій, можливість випуску такої кількості продукції, яка забезпечить проходження точки беззбитковості й отримання необхідних прибутків.

## **6.2 Розрахунок капітальних витрат на проведення проекту**

Усі засоби, які були використані для розробки сайту є безплатними та знаходяться у вільному доступі.

Нижче наведено перелік програмних засобів, що були використані для виконання поставленого завдання:

- SQL Server 2014 Management Studio;
- Браузер для відображення контенту;
- Visual Studio 2015;
- домашній комп'ютер або ноутбук;
- хмарна технологія Microsoft Azure.

Витрати на вище перерахуванні програмні засоби становлять 0 гривень. їх ділком достатньо для виконання поставленого завдання.

## **6.3 Визначення витрат на оплату праці працівників задіяних у проекті**

Досліджуване питання є вкрай актуальним, оскільки у дійсний час, при переході до ринкових відносин, основним мотивуючим чинником працівників є бажання мати гарантовану заробітну плату. При цьому ні інтенсивність, ні якість праці в розрахунок не беруться, переважає бажання мати спокійну роботу з невеликим, але гарантованим заробітком, ніж інтенсивну роботу з



високою оплатою. Оплата праці — чи не найважливіша категорія у системі умов праці.

В даному випадку потрібно розрахувати витрати на оплату праці працівників, що задіяні у розробці даного проекту. Для виконання поставленої задачі потрібні працівники. Величина заробітної плати повинна відповідати умовам при яких працівник буде працювати якомога продуктивніше, а підприємство отримає максимальний прибуток.

При розрахунку собівартості проекту враховується не тільки заробітна плата працівників, але і сума нарахувань, які нараховуються до заробітної плати кожного працівника. Часто на практиці для розрахунку затрат підприємства на заробітну плату працівників користуються узагальненим коефіцієнтом нарахувань до заробітної плати, який на сьогодні становить 36,77%.

Такий коефіцієнт узагальнює усі нарахування і служить для усереднених розрахунків витрат підприємства на нарахування до заробітної плати працівників. Для розрахунку нарахувань до заробітної плати можна використати наступну формулу:

$$NZp = Zp * kzp, \quad (3.1)$$

де  $NZp$  - сума нарахувань до заробітної плати;  $Zp$  - заробітна плата працівника;  $kzp$  — коефіцієнт нарахувань до заробітної плати.

Основна заробітна плата складається із прямої і доплати, яка при розрахунках становить 25% - 35% від прямої зарплати. При розрахунку з/п кількість робочих днів в місяці слід приймати - 25 дні/міс., що відповідає 200 год./міс.

Пряма зарплата визначається

$$Zp = (O_i * T_i) / 200, \quad (3.2)$$

де  $O_i$  - розмір місячних окладів  $i$ -х категорій працівників;  $T_i$  - трудомісткість робіт виконаних працівниками  $i$ -х категорій.

Отже, при залученні для реалізації проекту одного працівників з місячним окладом у сумі 3600 грн. В даному випадку на суму нарахованої заробітної плати проводять нарахування у розмірі 36,77 % необхідно що місяця затрачати на нарахування до заробітної плати працівників 1324 грн.

Сумарні витрати підприємства на заробітну плату враховують відповідні нарахування і визначаються за формулою:

$$SZp = Zp + NZp, \quad (3.3)$$

де  $SZp$  - сумарні витрати підприємства по заробітній платі працівників;  
 $Zp$  - заробітна плата працівників;  $NZp$  - сума нарахувань до заробітної плати;

Отже, протягом одного місяців для розробки програмного продукту для оплати праці одному працівникам необхідно затратити  $3600+1324=4924$  грн.

#### **6.4 Визначення суми амортизаційних відрахувань**

В основному величина амортизаційних відрахувань залежить від темпів виробництва і вартості обладнання, що при цьому використовується. Амортизаційні нарахування розраховуються за допомогою узагальненої формули з використанням коефіцієнту амортизації, який враховує інтенсивність використання обладнання, галузь в якій працює підприємство і інші фактори.

$$A = BL/Tkv, \quad (3.4)$$

де  $BL$  - балансова вартість обладнання, що використовується для реалізації проекту;  $Tkv$  — термін корисного використання .

При реалізації проекту по розробці програмного забезпечення буде використано 1 комп'ютери загальною балансовою вартістю 10000 грн. Таким чином, використовуючи залежність (3.4) розрахуємо суму амортизаційних нарахувань, яку потрібно затратити при реалізації запропонованого нововведення  $10000/2=5000$  грн., необхідно знайти амортизаційні відрахування

за 1 місяць  $5000/12=416,7$  грн. Визначаємо амортизаційні відрахування за 25 днів.

Отже, сума витрат на амортизаційні відрахування становить 416,7 грн.

### **6.5 Розрахунок вартості електроенергії, необхідної для реалізації проекту**

Величина споживання енергоносіїв не рідко є важливим показником у собівартості продукції і має важливий вплив на позитивне чи негативне рішення щодо впровадження нових проектних рішень. При цьому необхідно розрахувати затрати на електроенергію враховуючи два основних визначальних фактори - це сумарне споживання електроенергії обладнанням і ціна одиниці електроенергії. Враховуючи ці показники необхідно використати наступну формулу для розрахунку витрат коштів на електроенергію:

$$SE=SO*CE, \quad (3.5)$$

де  $BO$  - узагальнений показник споживання електроенергії обладнання, що використовується при реалізації проекту, кВт;  $CE$  - ціна одного кіловата електроенергії за чинними розцінками, грн.

За технічними характеристиками обладнання, що планується використати для реалізації запропонованого нововведення буде споживати по 0,15 кВт електроенергії за одну годину. За умовами реалізації проекту плануємо, що обладнання буде працювати повний робочий день, а саме 8 годин. За даних умов роботи для функціонування обладнання буде потрібно  $8*0,15 = 1,2$  кВт\*год електроенергії на один робочий день. Тоді на протязі одного місяця розробки проекту потрібно буде споживати  $1,2*25$  (днів) = 30 кВт\*год електроенергії. Ціна однієї кВт\*год електроенергії на сьогоднішній день для гуртожитків України становить 0,72 грн. Отже, використовуючи залежність (3.5) розрахуємо вартість електроенергії, що буде споживати обладнання в результаті реалізації проекту на протязі двох місяців:

$$SE = 30 * 0,72 = 21,6 \text{ грн} \quad (3.6)$$

Провівши відповідні розрахунки можна стверджувати, що для впровадження проекту на споживання електроенергії обладнанням необхідно витратити 21,6 грн.

### **6.6 Визначення витрат підприємства на сировину і матеріали пов'язані із реалізацією проектного рішення і узагальнення його собівартості**

Для визначення витрат сировини і матеріалів пов'язаних із проектом потрібно знати планову продуктивність, що стосується даного проекту (скільки розхідних необхідно затратити) при умові виконань даних робіт в потоці.

В таблиці 3.1 приведені затрати на сировину і матеріали, необхідні для повної реалізації роботи в процесі розробки програмного продукту.

Таблиця 3.1 - Затрати на сировину і матеріали

<b>№ п/п</b>	<b>Показник</b>	<b>Один. вимір.</b>	<b>Необхідна Кількість</b>	<b>Ціна одиниці, грн.</b>	<b>Сума, грн.</b>
1	Канцелярські матеріали		1	100	100
	Разом				100

Отже, сумарні затрати на сировину та матеріали в результаті впровадження проекту становлять 100 грн.

В багатьох випадках існує ряд додаткових витрат, які пов'язані із реалізацією проекту, але в більшості випадків їхня сума не перевищує десяти відсотків від загальної собівартості реалізації проекту. Тому при розрахунку загальної річної собівартості проекту додатково можна враховувати ще десять відсотків загальної собівартості в якості додаткових витрат.

Після проведених розрахунків основних статей затрат що стосуються реалізації проекту нижче наведено їх узагальнення і подано їх список в табличній формі (див. табл. 3.2).

Таблиця 3.2 - Розрахунок собівартості програмного продукту

№ п/п	Показник	Сума, грн.
1	Затрати на сировину і матеріали	100
2	Амортизаційні відрахування	417
3	Витрати на електроенергію	22
4	Витрати на оплату праці	4924
5	Сума	5463
6	Інші додаткові витрати	92
	Разом	5555

Отже, провівши розрахунок загальної собівартості реалізації запропонованого нововведення можна зробити висновок, що для реалізації проекту необхідні кошти у сумі 5555 грн.

### 6.7 Розрахунок вартості програмного продукту

Ціна складається з декількох компонентів:

$$Ц = C + 17 + ПДВ \quad (3.7)$$

де  $C$  - собівартість програмного продукту,  $\Pi$  - прибуток, який становить 40% від собівартості  $\Pi = 5555 * 0,4 = 2222$  грн., ПДВ - податок на додану вартість, який береться в розмірі 20% від суми собівартості і прибутку.  $\underline{ПДВ} = 5555 + 2222 * 0,2 = 1555$  (грн.)

Підставляючи значення у формулу, ціна програмного продукту становить:  $Ц = 5555 + 2222 + 1555 = 9332$  (грн.)

## 6.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

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

(3.8)

$$E_p = \frac{\Pi}{C_v}$$

де  $\Pi$  - прибуток,  $C_v$  - собівартість.

Підставляючи значення у формулу, економічна ефективність програмного продукту становить:  $E_p = 2222/5555 = 0,4$ .

Термін окупності програмного продукту:

$$T_p = 1/E_p. \quad (3.9)$$

Отже,  $T_p = 1/0,4 = 2,5$  років. Зведені техніко-економічні показники програмного продукту (див табл. 3.3).

Таблиця 3.3 - Техніко-економічні показники програмного продукту

№ п/п	Показник	Значення
1	Собівартість, грн.	5555
2	Плановий прибуток, грн.	2222
3	Ціна, грн.	7777
4	Економічна ефективність	0,4
5	Термін окупності, років	2,5

В ході розрахунків було визначено, що показник економічної ефективності становить 0,4, а також термін окупності — 2,5 року. Зважаючи на те, що нормативним показником терміну окупності є показник до 5 років, то можна зробити висновок, що розробка системи є в межах високої ефективності.

### **Висновок до розділу 6**

Розробка Інтернет-магазину побутової техніки «Мрія» з використанням «хмарної» платформи Microsoft Azure є ефективною. У даній розробці усі програмні продукти є безкоштовними, витрати на заробітною платню розраховані правильно, розрахунок електроенергії, амортизаційних відрахувань, затрати на сировину та матеріали є не значними. Тому усі ці показники значно підвищили економічну ефективність і термін окупності даної розробки.

## **7. Охорона праці та безпека в надзвичайних ситуаціях**

### **7.1 Охорона праці**

#### **7.1.1 Захист користувачів ПК від різного типу випромінювання**

Всесвітня організація охорони здоров'я ще в 1989 р. в офсетній публікації N 99 дійшла висновку, що робота з використанням персональних комп'ютерів супроводжується зоровим і нервово-емоційним напруженням, негативними зрушеннями в кістково-м'язовій системі людини. Слабкі рівні неіонізуючих й іонізуючих випромінювань, що створюються відео дисплейними терміналами (ВДТ) на електронно-променевих трубках, несуть загрозу збільшення онкопатологій, негативного впливу на вагітних і плід. У всіх розвинених країнах, у тому числі в країнах Європейської спільноти, існують сотні документів, які регламентують вимоги не тільки до комп'ютерів, а й до організації робочих місць з їх використанням. Таким чином, безконтрольне використання комп'ютерної техніки може призвести до негативного впливу на здоров'я користувачів комп'ютерів, особливо дітей.

У сучасних умовах важко знайти галузь інтелектуальної праці, де б не використовувалися комп'ютери. Новітній навчальний процес у загальноосвітній та вищій школі також не можна уявити без комп'ютерів. Разом з тим проблеми охорони праці тих, хто працює за комп'ютером, не піддаються активному дослідженню. Органи управління охороною праці та Міністерство охорони здоров'я України вивчають умови праці, світову практику і видають нормативно-правові акти, присвячені охороні праці осіб, які працюють на електронно-обчислювальних та обчислювальних машинах, але застосування їх на практиці викликає безліч спірних питань. У Союзі РСР діяли «Тимчасові санітарні норми і правила для працівників обчислювальних центрів», затверджені заступником.

Головного державного санітарного лікаря СРСР 2 березня 1988 р. (№ 4559-88). У цьому акті йшлося про кількість опрацьованих символів (знаків), про додаткові перерви (тривалістю 5-10 хвилин) для відпочинку та інші вимоги до приміщень та організації робочих місць. Але в цьому акті зазначалося, що у разі виконання у повному обсязі рекомендованих комплексних оздоровчих



заходів, вимог до вентиляції, рівнів шуму і вібрації, освітлення, захисту від статичної електрики й випромінювання, правильної організації робочого місця в приміщенні роботи працівників (у тому числі і тих, хто працює на комп'ютерах), належить до I-II класу за гігієнічними умовами праці, тобто до оптимальних або допустимих (гігієнічна класифікація праці Мінохорони здоров'я СРСР № 4137- 86). На підставі Міністерство праці та соціальної політики зробило висновок про те, що умови праці на комп'ютерах не належать до класу шкідливих, і ті, хто працює на них, пільгами і компенсаціями за умови праці (у тому числі і додатковою відпусткою) не користуються.

Згодом в Україні були прийняті нові нормативно-правові акти щодо охорони праці тих, хто працює за комп'ютером. До них належать «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПНЗ.3.2.007-98. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Перш за все, зазначені правила передбачають, що особи, які працюють з ВДТ ЕОМ і ПЕОМ, підлягають обов'язковим медичним оглядам попереднім - при влаштуванні на роботу та періодичним - протягом трудової діяльності. Періодичні медичні огляди мають проводитися раз на два роки комісією в складі терапевта, невропатолога та офтальмолога. Основними критеріями оцінки придатності до роботи з ВДТ ЕОМ і ПЕОМ мають бути показники стану органів зору: гострота зору, показники рефракції; стану біокулярного апарату ока тощо. При цьому необхідно враховувати також стан організму в цілому. Кожні два роки жінки, що працюють з ВДТ ЕОМ та ПЕОМ, обов'язково оглядаються лікарем акушером-гінекологом. У цих правилах передбачено й протипоказання для роботи за комп'ютерами. Це усі хронічні форми психічних захворювань, ендокринні захворювання, тяжкий ступінь бронхіальної системи, порушення менструальної функції, що супроводжується матковими кровотечами, гіпертонічна хвороба III стадії.

Площа, виділена для одного робочого місця з відео терміналом або персональною ЕОМ, має бути не меншою 6 м<sup>2</sup>, а обсяг - не менше 20м<sup>3</sup>. Робоче

сидіння користувача повинно бути підйомно-поворотним, таким, що регулюється за висотою, кутом нахилу сидіння та спинки, за відстанню спинки до переднього краю сидіння, висотою підлокітників.

По-друге, правила передбачають, що для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності слід надавати їм перерви для відпочинку. Виділено три професійні групи: інженери-програмісти, оператори електронно-обчислювальних та обчислювальних машин й оператори комп'ютерного набору. Усім цим працівникам слід надавати перерви для відпочинку. Розробники програм (інженери-програмісти) виконують роботу переважно з відео терміналом та документацією за необхідності інтенсивного обміну інформацією з ЕОМ і високою частотою прийняття рішень. Робота характеризується інтенсивною творчою розумовою працею з підвищеним напруженням зору, концентрацією уваги на фоні нервово-емоційного напруження, вимушеною робочою позою, загальною гіподинамією, періодичним навантаженням на кисті верхніх кінцівок. Робота виконується в режимі діалогу з ЕОМ у вільному темпі з періодичним пошуком помилок в умовах дефіциту часу.

Встановлюються такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при восьмигодинній денній робочій зміні залежно від характеру праці:

- для розробників програм із застосуванням ЕОМ слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;

- для операторів із застосуванням ЕОМ слід призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;

- для операторів комп'ютерного набору слід призначати регламентовані перерви для відпочинку тривалістю 10 хвилин після кожної години роботи за ВДТ.

У періоди, що передують появі об'єктивних і суб'єктивних ознак стомлення і зниження працездатності можна надавати додаткові перерви для відпочинку.

Детально питання про надання таких перерв можна вирішувати у колективних договорах. Перерви, як свідчить практика, працівники використовують, і скарг на відмову у наданні таких перерв немає. Складніше вирішується питання про надання додаткових відпусток за несприятливі умови і праці.

17 листопада 1997 р. Кабінетом Міністрів України було затверджені Списки виробництв, робіт, цехів, професій і посад, зайнятість працівників в яких дає право на щорічні додаткові відпустки за роботу із шкідливими і важкими умовами праці та за особливий характер праці. У Додатку № 2 було передбачено право оператора електронно-обчислювальних машин (позиція 43) на додаткову з відпусток тривалістю чотири календарні дні за особливі умови праці.

## **7.2 Безпека в надзвичайних ситуаціях**

### **7.2.1 Особливості роботи та розлади здоров'я користувачів комп'ютерів, що формуються під впливом роботи за комп'ютером**

У професійних операторів частіше зустрічаються порушення органів зору, опорно-рухового апарату, центральної нервової, серцево-судинної, імунної та статеві систем, захворювання шкіри. Зафіксована значна кількість скарг операторського персоналу на загальне недомагання, передчасне стомлювання, головний біль, порушення функцій органів зору, які здійснювали несприятливий психофізіологічний вплив на самопочуття та працездатність операторів.

Сучасна професія користувача ВДТ належить до розумової праці, яка характеризується: високою напруженістю зорових функцій; одноманітною позою; великою кількістю стереотипних висококоординованих рухів, що виконуються лише м'язами кистей рук на фоні малої загальної рухової активності; значним нервово-емоційним компонентом, особливо в умовах

дефіциту часу; роботою з великими масивами інформації, що викликає активізацію уваги та інших вищих психічних функцій. Крім того, при роботі з дисплеями на електронно-променевих трубках виникає вплив на користувача цілої низки факторів фізичної природи - електростатичні поля, радіочастотне та рентгенівське випромінювання тощо.

Діяльність професіоналів можна поділити на три групи:

-Діяльність, яка пов'язана із здійсненням логічних операцій, що постійно-повторюються. Це робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва

-Діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження.

-Діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму.

У користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень досить часто виникають психологічні та поведінкові порушення. Серед користувачів ВДТ значного поширення набуло специфічне захворювання, яке отримало назву синдром комп'ютерного стресу. СКС супроводжується головним болем, запаленням очей, алергією, роздратованістю. Інформаційне перевантаження користувачів ВДТ супроводжується низкою специфічних захворювань, які називають інформаційними. Першим симптомом їх є головний біль. Дослідження, проведені в США, Швейцарії та інших країнах, показали, що робота з обслуговування ВДТ супроводжується підвищеним напруженням зору, інтенсивністю і монотонністю праці, збільшенням статичних навантажень, нервово-психічним напруженням, впливом різного виду випромінювань. Внаслідок цього серед операторів ВДТ, як зазначають фахівці всесвітньої організації охорони здоров'я, частіше, ніж в інших групах працюючих, трапляються такі професійні захворювання, як передчасна стомлюваність, погіршення зору, м'язові і головні болі, психічні й нервові розлади, хвороби серцево-судинної системи, онкологічні захворювання. Вважається, що стан організму операторів ВДТ визначається комплексним впливом факторів

трудового процесу і середовища, значення яких є неоднаковим. На операторів з малим стажем роботи на ВДТ домінуючий вплив чинять фактори середовища, а на операторів зі стажем понад 5 років — фактори трудового процесу.

Комп'ютерний зоровий синдром (КЗС) - комплекс порушень здоров'я, який може виникати у користувачів персональних комп'ютерів (ПК). Діагноз ставлять, якщо людина, що працює за ПК протягом двох годин, висловлює хоча б дві з десяти скарг:

- свербіж;
- важкість в очах;
- головний біль;
- сльозотеча;
- різь;
- туман;
- двоїння;
- фотофобія;
- миготіння знаків на екрані;
- нудота.

У користувачів ПК дуже поширені кон'юнктивіти і блефарити, патогенетично пов'язані з КЗС.

Синдром розвивається при умові, що робоче місце організовано неправильно - у користувача незручне крісло, відсутні попітри для паперів, підставки для ніг та кистей рук, не встановлена висота і нахил монітора відносно очей, відстань від очей до екрана. За таких умов тіло людини при роботі займає вимушене положення: спина статично напружена, шия витягнута, плечі жорстко фіксовані. Напружені м'язи погіршують кровотік у сонних артеріях, а недостатнє кровозабезпечення головного мозку веде до очманіння, появи головного болю. На фоні шийного остеохондрозу з'являється відчуття випирання очних яблук, туману в очах, мушок та райдужних кіл у полі зору. Розвитку КЗС сприяє поганий мікроклімат приміщення, значна загальна іонізація та мікробне забруднення.

У операторів ВДТ "очні" симптоми трапляються частіше, ніж "зорові", причому частота проявів астенії вища у жінок, ніж у чоловіків і більше виражена в осіб середнього і старшого віку. Причиною вважається електромагнітне випромінювання від ВДТ.

Робота з ВДТ може призвести до розвитку короткозорості, так як у користувачів комп'ютерів, в основному, "працює" ближній зір.

При аналізі зорової роботи операторів ВДТ, встановлено, що через дві години частота флуктуацій акомодатії зменшується, а внесок низькочастотної компоненти підвищується. Це може бути причиною скарг на втому зорового аналізатора. Тривала робота на ВДТ може призвести до розвитку короткозорості, оскільки у користувачів ВДТ головним чином "працює" ближній зір.

Робота за комп'ютером характеризується також тим, що постійний напружений погляд на екран монітора зменшує частоту моргання. При цьому погіршується зволоження поверхні очного яблука слезовою рідиною, яка захищає рогівку ока від висихання, пилу та інших забруднень. Це може призвести до виникнення так званого синдрому Сітка: рогівка висихає і мутніє, і як наслідок розвивається сліпота.

Отже, порушення зорових функцій користувачів ВДТ пов'язані, головним чином, з чотирма групами факторів:

- параметрами освітлення робочого місця;
- характеристиками дисплея;
- неправильною організацією робочого місця;
- специфікою роботи на ВДТ;
- виробництва і вартості обладнання, що при цьому використовується.

### **7.2.2 Організації робочого місця користувача відео дисплейним терміналом**

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей

трудової діяльності (ГОСТ 12.2.032-78, ГОСТ 22.269-76, ГОСТ 21.889-76). конструкція робочого місця користувача ВДТ має забезпечити підтримання оптимальної робочої пози. Робочі місця з ВДТ слід так розташовувати відносно світлових прорізів, щоб природне світло падало збоку, переважно зліва.

При розміщенні робочих столів з ВДТ потрібно дотримуватись таких відстаней: між бічними поверхнями ВДТ - 1,2 м; від тильної поверхні одного ВДТ до екрана - 2,5 м. Екран ВДТ має розташовуватися на оптимальній відстані від очей користувача, що становить 600-700 мм, але не ближче ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів. Розташування екрана ВДТ має забезпечувати зручність зорового спостереження у вертикальній площині під кутом  $+30^{\circ}$  до нормальної лінії погляду працюючого.

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідно застосовувати приєкранні фільтри, локальні світлофільтри та інші засоби захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат. При оснащенні робочого місця з ВДТ лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам ДСанПіН 3.3.2.007-98.

При організації праці, пов'язаної з використанням ВДТ ЕОМ і ПЕОМ, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності передбачаються внутрішньозмінні регламентовані перерви для відпочинку. Внутрішньозмінні режими праці і відпочинку містять додаткові нетривалі перерви в періоди, що передують появі об'єктивних і суб'єктивних ознак стомлення і зниження працездатності. При виконанні робіт, що належать до різних видів трудової діяльності, за основну роботу з ВДТ слід вважати таку, що займає не менше 50% робочого часу. Впродовж робочої зміни мають передбачатися:

- перерви для відпочинку і вживання їжі;
- перерви для відпочинку і особистих потреб;
- додаткові перерви, що вводяться для окремих професій з урахуванням особливостей трудової діяльності.

За характером трудової діяльності розрізняють три професійні групи, згідно з діючим класифікатором професій:

-розробники програм виконують роботу переважно з відеотерміналом та документацією при необхідності інтенсивного обміну інформацією з ЕОМ і високою частотою прийняття рішень. Робота характеризується інтенсивною розумовою творчою працею з підвищеним напруженням зору, концентрацією уваги на фоні нервово-емоційного напруження, вимушеною робочою позою, загальною гіподинамією, періодичним навантаженням на кисті верхніх кінцівок. Робота виконується в режимі діалогу з ЕОМ у вільному темпі з періодичним пошуком помилок в умовах дефіциту часу;

-оператори електронно-обчислювальних машин виконують роботу, пов'язану з обліком інформації, одержаної з ВДТ за попереднім запитом, супроводжується перервами різної тривалості, пов'язана з виконанням іншої роботи і характеризується напруженням зору, невеликими фізичними зусиллями, нервовим напруженням середнього ступеня та виконується у вільному темпі;

-оператор комп'ютерного набору виконує одноманітні за характером роботи з документацією та клавіатурою і нечастими нетривалими переключеннями погляду на екран дисплея, з введенням даних з високою швидкістю. Робота характеризується як фізична праця з підвищеним навантаженням на кисті верхніх кінцівок на фоні загальної гіподинамії з напруженням зору (фіксація зору переважно на документи), нервово-емоційним напруженням.

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 години. При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин. Для зниження нервово-емоційного напруження, втомлення зорового аналізатора, поліпшення мозкового кровообігу, подолання несприятливих наслідків



гіподинамії, запобігання втомі доцільно деякі перерви використовувати для виконання комплексу вправ.

### **Висновок до розділу 7**

В даному розділі наведено методи захисту користувачів ПК від типу випромінювання, наведені правила використання для зменшення шкоди здоров'ю, було розглянуто особливості роботи та розлади здоров'я користувачів комп'ютерів, наведено різноманітні захворювання, які можуть спричинити довготривале праця за комп'ютером і наведено правила, ГОСТи для організації робочого місця користувача відео дисплейним терміналом .

## **8. Екологія**

### **8.1 Джерела шуму і вібрацій, методи їх знешкодження**

Джерела шуму і вібрації - є машини і механізми з неврівноваженими обертовими масами, а також технологічні установки і апарати, в яких рух газів і рідин відбувається з великими швидкостями і має пульсуючий характер. Сучасний розвиток техніки, оснащення підприємств потужними і швидкими машинами і механізмами призводить до того, що людина постійно піддається впливу шуму все зростаючій інтенсивності. Підвищення рівня шуму та вібрації на робочих місцях надає шкідливий вплив на організм людини. В результаті тривалого впливу шуму порушується нормальна діяльність серцево-судинної і нервової системи, травних і кровотворних органів, розвивається професійна приглухуватість, прогресування якої може привести до повної втрати слуху.

Шум та вібрація різняться між собою частотою коливань. Джерелом шуму є будь-який процес, що викликає місцеву зміну тиску або механічні коливання в жорстких, водянистих або газоподібних середовищах. Джерелами шуму можуть бути двигунки, насоси, компресори, турбіни, пневматичні і електричні інструменти, молоти, долота, верстати, центрифуги, бункери та інші установки, що мають пересуваються деталі. Не рахуючи того, за останні роки у зв'язку зі значимим розвитком міського транспорту зросла інтенсивність шуму й у побуті, тому як несприятливий фактор він придбав величезне соціальне значення.

#### **8.1.1 Вплив шуму на організм людини**

Реакція людини на шум різна. Деякі люди терпимі до шуму, у інших він викликає роздратування, прагнення піти від джерела шуму. Психологічна оцінка шуму в основному базується на понятті сприйняття, причому велике значення має внутрішня налаштування до джерела шуму. Вона визначає, чи буде шум сприйматися як заважає. Часто шум, відтворений самою людиною, не турбує його, в той час як невеликий шум, викликаний сусідами або яким-небудь іншим джерелом, надає сильний подразнюючий ефект.

В умовах сильного міського шуму відбувається постійна напруга слухового аналізатора. Це викликає збільшення порогу чутності (10 дБ для більшості людей з нормальним слухом) на 10-25 дБ. Шум ускладнює розбірливість мовлення, особливо при його рівні більше 70 дБ. Збиток, який заподіює слуху сильний шум, залежить від спектра звукових коливань і характеру їх зміни. Небезпека можливої втрати слуху через шуму в значній мірі залежить від індивідуальних особливостей людини. Постійний вплив сильного шуму може не тільки негативно вплинути на слух, але й викликати інші шкідливі наслідки - дзвін у вухах, запаморочення, головний біль, підвищену втому.

Надмірний шум може стати причиною нервового виснаження, психічної пригніченості, вегетативного неврозу, виразкової хвороби, розлади ендокринної та серцево-судинної систем. Шум заважає людям працювати і відпочивати та знижує продуктивність праці.

Найбільш чутливі до дії шуму особи старшого віку. Так, у віці до 27 років на шум реагують 46% людей, у віці 28-37 років - 57%, у віці 38-57 років - 62%, а у віці 58 років і старше - 72%.

Високі рівні шуму в міському середовищі, що є одним з агресивних подразників центральної нервової системи, здатні викликати її перенапруження. Міський шум справляє негативний вплив і на серцево-судинну систему.

### **8.1.2 Методи і засоби захисту від шуму**

Одним із напрямків боротьби з шумом є розробка державних стандартів на засоби пересування, інженерне обладнання, побутові прилади, в основу яких покладені гігієнічні вимоги щодо забезпечення акустичного комфорту.

Санітарні норми допустимого шуму обумовлюють необхідність розробки технічних, архітектурно-планувальних та адміністративних заходів, спрямованих на створення відповідає гігієнічним вимогам шумового режиму, як в міській забудові, так і в будівлях різного призначення, дозволяють зберегти здоров'я та працездатність населення. Одним з ефективних засобів боротьби з

виробничим шумом є використання демпфуючих металевих і неметалічних матеріалів. Однак неметали не використовуються для зниження шуму зіткнень з-за їх невисоких характеристик міцності, а металеві матеріали, що характеризуються високими міцними властивостями, забезпечують зниження шуму досить незначно, тому постало питання про створення принципово нових матеріалів, які могли б мати високі міцні характеристики і достатні демпфуючі властивості. Такими матеріалами є біметали, які дозволяють отримувати таке поєднання службових властивостей, яке не можна отримати в одному окремо взятому металі або сплаві, наприклад: високу міцність з корозійною стійкістю, ударну в'язкість з зносостійкістю, міцність з високою електро-і теплопровідністю, високу міцність і достатні демпфуючі властивості. До цих пір боязкі спроби використовувати біметали для зниження шуму і вібрації не забезпечили вирішення проблеми, тому досить актуальним є наукове дослідження, присвячене розробці біметалів з підвищеними демпфуючими властивостями. Технічні засоби захисту від шуму: звукопоглинання, звукоізоляція, екранування, кошти демпфування і глушники шуму.

Заходи боротьби з шумом:

- заміна гучних процесів безшумними або менш гучними;
- поліпшення якості виготовлення і монтажу обладнання;
- укриття джерел шуму;
- висновок працюючих людей зі сфери шуму;
- застосування індивідуальних захисних засобів.

### **8.1.3 Дія вібрації на організм людини**

При роботі в умовах вібрацій продуктивність праці знижується, зростає кількість травм. Не завжди відповідають нормам рівні вібрацій на органах управління. Зазвичай у спектрі вібрації переважають низькочастотні вібрації негативно діють на організм. Деякі види вібрації несприятливо впливають на нервову і серцево-судинну системи, вестибулярний апарат. Найбільш шкідливий вплив на організм людини надає вібрація, частота якої збігається з частотою власних коливань окремих органів, приблизні значення яких такі

(Гц): шлунок - 2-3; нирки - 6-8; серце - 4-6 ; кишечник - 2-4; вестибулярний апарат - 0,5-3; очі - 40-100.

Локальна вібрація вражає нервово-м'язові тканини і опорно-руховий апарат і призводить до спазмів периферичних судин. При тривалих і інтенсивних вібраціях в деяких випадках розвивається професійна патологія (до неї частіше призводить локальна вібрація): периферична, церебральна або церебрально-периферична вібраційна хвороба. В останньому випадку спостерігаються зміни серцевої діяльності, загальне збудження або, навпаки, гальмування, стомлення, поява болю, відчуття трясіння внутрішніх органів, нудота. У цих випадках вібрації впливають і на кістково-суглобовий апарат, м'язи, периферійний кровообіг, зір, слух. Місцеві вібрації викликають спазми судин, які розвиваються з кінцевих фаланг пальців, поширюючись на всю кисть, передпліччя, і охоплюють судини серця.

Максимальна амплітуда коливань черевної стінки спостерігається на частотах 7-8 Гц. Зі збільшенням частоти коливань їх амплітуда при передачі по тілу людини послаблюється. У положенні стоячи і сидючи ці ослаблення на кістках тазу рівні 9 дБ на октаву зміни частоти, на грудях і голові - 12дБ, на плечі -12-14 дБ. Ці дані не поширюються на резонансні частоти, при дії яких відбувається не послаблення, а збільшення швидкості коливань.

У виробничих умовах ручні машини, вібрація яких має максимальні рівні енергії у смугах низьких частот викликають вібраційну патологію з переважним ураженням нервово-м'язової тканини та опорно-рухового апарату.

При роботі з ручними машинами, вібрація яких має максимальний рівень енергії у високочастотній області спектру (вище 125 Гц), виникають головним чином судинні розлади. При впливі вібрації низької частоти захворювання виникає через 8-10 років, а при дії високочастотної вібрації - через 5 років і раніше. Загальна вібрація різних параметром викликає різну ступінь вираженості змін нервово й системи, серцево- судинної системи і вестибулярного апарату.

Виробнича вібрація, що характеризується значною амплітудою і тривалістю дії, викликає у працюючих дратівливість, безсоння, головний біль,

ниючий біль в руках людей, що мають справу з вібруючим інструментом. При тривалому впливі вібрації перебудовується кісткова тканина: на рентгенограмах можна помітити смуги, схожі на сліди перелому - ділянки найбільшої напруги, де розм'якшується кісткова тканина. Зростає проникність дрібних кровоносних судин, порушується нервова регуляція, змінюється чутливість шкіри. При роботі з ручним механізованим інструментом може виникнути акроасфіксія - втрата чутливості, побіління пальців, кистей рук. При дії загальної вібрації більш виражені зміни з боку центральної нервової системи: з'являються запаморочення, шум у вухах, погіршення пам'яті, порушення координації рухів, вестибулярні розлади, схуднення. Якщо вібрація діє в горизонтальній площині по осі, перпендикулярній хребту, то резонансна частота тіла обумовлена згинанням хребта і жорсткістю тазостегнових суглобів. Область резонансу для голови сидячої людини відповідає 20-30 Гц. У цьому діапазоні амплітуда віброприскорення голови може втричі перевищувати амплітуду коливань плечей. Якість зорового сприйняття предметів значно погіршується при частоті 60-70 Гц, що відповідає резонансу очних яблук.

#### **8.1.4 Методи і засоби захисту від вібрації**

Для захисту від вібрації застосовують такі методи: зниження віброактивності машин; відбудова від резонансних частот; вібродемпфірування; віброізоляція; віброгасіння, а також індивідуальні засоби захисту.

Зниження віброактивності машин (зменшення Біті) досягається зміною технологічного процесу, застосуванням машин з такими кінематичними схемами, при яких динамічні процеси, викликані ударами, прискореннями і були б виключені або гранично знижені.

Віброізоляція полягає у зменшенні передачі коливань від джерела до захищається за допомогою пристроїв, які розміщені між ними. Для віброізоляції найчастіше застосовують віброізолюючі опори типу пружних прокладок, пружин або їх поєднання. Ефективність віброізоляторів оцінюють коефіцієнтом передачі КП, рівним відношенню амплітуди вібропереміщення,

віброшвидкості, віброприскорення, що захищається, або діє на нього сили до відповідного параметру джерела вібрації. Віброізоляція тільки в тому випадку знижує вібрацію, коли  $KП < 1$ . Чим менше КП, тим ефективніше віброізоляція.

Зменшення вібрації в джерелі виникнення досягають зміною технологічного процесу з виготовленням деталей з капрону, гуми, текстоліту, своєчасним проведенням профілактичних заходів та мастильних операцій; центруванням і балансуванням деталей; зменшенням зазорів у з'єднаннях. Передачу коливань на підставу агрегату або конструкцію будинку послаблюють допомогою екранування, що є одночасно засобом боротьби і з шумом.

Для підвищення захисних властивостей організму, працездатності і трудової активності слід використовувати спеціальні комплекси виробничої гімнастики, вітамінну профілактику (два рази на рік комплекс вітамінів С, В, нікотинову кислоту), спецхарчування.

## **Висновок до розділу 8**

В даному розділі наведено джерела шуму і вібрацій, методи їх знешкодження. Було наведено методи і засоби захисту, визначено правила, зменшення рівня шуму і вібрації, наведено дії вібрації і шуму на організм людини.

## Висновки

В даній роботі було проаналізовано Інтернет-магазину побутової техніки на «хмарній» технології Microsoft Azure. В даній роботі використовувалася студентська підписка від DreamSpark. Дана підписка дозволяє автоматично будувати, тестувати і розгорнути веб-додатки, а також створювати MySQL.

На даний момент йде активна розробка і вдосконалення технології «хмарних обчислень». Переваги від впровадження даної технології очевидні. Адже це економія для споживачів, боротьба з піратством для розробників, мінімізація витрат в ІТ сфері для бізнесу, уніфікація мережевих стандартів для всіх користувачів. Також важливо відзначити, що «хмара» з тисяч машин здатна вирішувати дуже важкі завдання, які необхідні сучасним ученим всіх галузей. Так що технології «хмарних обчислень» є перспективною складовою ІТ сфери, що швидко розвивається .

В даній роботі було описано дії, розміщено і проаналізовано веб-додаток на «хмарі», сервіси моніторингу для веб-додатків і представлено балансувальних LoadMaster від Microsoft Azure.

При балансуванні навантаження «хмарних» додатків слід враховувати особливості роботи додатків, особливості інфраструктури обчислювальних комплексів і додаткових вимог при організації сервісів:

- Неоднорідність структури розподіленого сервісу, різні запити вимагають різних обчислювальних потужностей і звернення до різних виконуючих модулів;

- Необхідність підтримувати безперервність довгострокової сесії користувача, тобто закріплення клієнта за конкретним з однотипних серверів на час роботи з сайтом;

- Неоднорідність структури обчислювального комплексу з точки зору продуктивності обчислювальних вузлів, що виконують однакові функції;

- Неоднорідність структури обчислювального комплексу з точки зору апаратних і програмних платформ обслуговуючих вузлів;



-Необхідність забезпечення відмовостійкості компонентів сервісу;

-Неоднорідність характеристик пропускної здатності лінії зв'язку, що з'єднують однотипні обчислювальні вузли (всі вузли в одній локальній мережі чи географічно розподілені).

-Можливість автоматичного масштабування додатків.

## Перелік посилань

1. Gillam, Lee. Cloud Computing: Principles, Systems and Applications / Nick Antonopoulos, Lee Gillam. — L.: Springer, 2010. — 379 с.

2. Риз Дж., Облачные вычисления: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 288 с.: ил

3. Стив Александер. Масштабируемость / Стив Александер // - Режим доступа: <http://citforum.ck.ua/hardware/articles/mashtab.shtml> - Дата доступа : 03.02.2017

4. Горизонтальное и вертикальное масштабирование. Взгляд со стороны бизнес приложений. - Режим доступа: <http://blog.vadmin.ru/2013/06/blog-post.html> - Дата доступа : 07.02.2017

5. Масштабирование нагрузки web-приложений. - Режим доступа: <http://habrahabr.ru/post/113992/> - Дата доступа : 07.01.2017

6. Системы балансировки нагрузки Web-серверов. - Режим доступа: <http://citforum.ru/internet/webservers/websbal.shtml> - Дата доступа : 15.01.2017

7. Балансировка нагрузки в распределенных системах. - Режим доступа: <http://www.intuit.ru/studies/courses/1146/238/lecture/6153> - Дата доступа : 27.01.2017

8. Балансировка в облаках. - Режим доступа: <http://www.osp.ru/os/2011/09/13011562/> - Дата доступа : 02.02.2017

9. Балансировка нагрузки в распределенных системах. - Режим доступа: <http://www.intuit.ru/studies/courses/1146/238/lecture/6153?page=2> - Дата доступа : 23.01.2017

10. Балансировка нагрузки: основные алгоритмы и методы. - Режим доступа: <http://habrahabr.ru/company/selectel/blog/250201/> - Дата доступа : 28.01.2017

11. DNS балансировка. - Режим доступа: <http://ruhighload.com/post/DNS> - Дата доступа : 03.02.2017

12. Балансировка масштабируемых приложений. - Режим доступа: <http://www.osp.ru/os/2012/08/13019244/> - Дата доступа : 03.02.2017

13. Load Balancing - Compute Engine - Google Cloud Platform. - Режим доступа: <https://cloud.google.com/compute/docs/load-balancing/> - Дата доступа : 04.02.2017

14. LoadMaster For Azure.

Режим доступа: <http://kemptechnologies.com/solutions/microsoft-load-balancing/loadmaster-azure/> - Дата доступа : 04.02.2017

15. AWS | Elastic Load Balancing- сетевой балансировщик нагрузки в облаке. - Режим доступа: <http://aws.amazon.com/ru/elasticloadbalancing/> - Дата доступа : 04.02.2017

16. Обзор балансировщиков нагрузки для \*nix. - Режим доступа: <https://хакер.ru/2014/03/15/62207/> - Дата доступа : 05.02.2017

17. LVS Introduction - Load Balancing Server Cluster. — Режим доступа: <http://www.linuxvirtualserver.org/whatis.html> - Дата доступа : 05.02.2017

18. Load Balancing - High Availability and Performance - NGINX. - Режим доступа: <http://nginx.com/solutions/load-balancing/> - Дата доступа : 10.02.2015

19. CloudStack: Configure Load Balancer. - Режим доступа: <https://support.getcloudservices.com/entries/22002577-CloudStack-Configure-Load-Balancer> - Дата доступа : 25.01.2017

20. SNMP. - Режим доступа: <http://xgu.ru/wiki/SNMP> - Дата доступа : 26.01.2017

