**Реферат**

**на тему:**

**Основні характеристики та задачі розподілених інформаційних систем.**

**Студентка 1-2з групи**

**Титаренко Маргарита**

**Станіславівна**

**Зміст**

[Вступ 3](#_Toc102042674)

[Визначення розподіленої інформаційної системи. 3](#_Toc102042675)

[Характеристика розподілених інформаційних систем. 4](#_Toc102042676)

[Прозорість 8](#_Toc102042677)

[Гнучкість 8](#_Toc102042678)

[Відкритість 9](#_Toc102042679)

[Масштабування 9](#_Toc102042680)

[Проблеми масштабування 10](#_Toc102042681)

[Способи масштабування 10](#_Toc102042682)

[Технології масштабування 11](#_Toc102042683)

[Висновки 14](#_Toc102042684)

[Список використаних джерел 14](#_Toc102042685)

Вступ

Сучасний світ просто немислимий без використання розподілених систем. Навіть у найпростішого мобільного додатка є API, через який підключається до хмарного сховища.

Винахід високошвидкісних мереж дозволило збирати комп'ютерні системи великої розмірності, які називаються розподіленими системами (РС), і призвело до швидкого зростання кількості додатків, що потребують розподіленої обробки. Зниження цін на системи зберігання даних, зростання пропускної спроможності мереж, а також кваліфікації користувачів дозволили зробити розподілені системи продуктивними та стійкими до відмови. Ці фактори, а також можливості бездротових та мобільних технологій зумовили швидкий розвиток розподілених додатків та інтерес до сфери розподілених систем з боку комп'ютерної спільноти, наукових, комерційних та урядових організацій. Таким чином, побудова розподілених обчислювальних систем — важлива сфера теоретичної та практичної інформатики, яка зачіпає всі аспекти розподілених обчислень та інформаційного доступу та потребує добре продуманого теоретичного обґрунтування.

Визначення розподіленої інформаційної системи.

Під розподіленою системою зазвичай розуміють систему, в якій апаратні та програмні компоненти, розташовані в мережевих комп'ютерах, взаємодіють та координують свої дії тільки шляхом надсилання повідомлень. Це найпростіше визначення веде до наступних

основним важливим особливостям розподілених систем: наявність у них паралельних процесів, що часто поділяють ресурси, відсутність глобальних годинників та незалежність відмов окремих компонентів. У літературі відсутнє канонічне визначення те, що є розподіленою системою. Визначення різних авторів не узгоджуються. Наприклад, в одній із книг сказано, що «під розподіленою системою ми розуміємо будь-яку обчислювальну систему,

у якій кілька комп'ютерів чи процесорів так чи інакше вступають у взаємодію». Під це визначення поруч із глобальними комунікаційними обчислювальними мережами потрапляють локальні мережі, багатопроцесорні комп'ютери, у яких кожен комп'ютер наділений власним пристроєм управління, і навіть системи взаємодіючих процесів.

Можна також вважати, що розподілена система є набір незалежних вузлів, що взаємодіють для вирішення проблеми, яка не може бути вирішена індивідуально.

При цьому важливо, що розподілена система є користувачем єдиною об'єднаною системою.

Моделью розподіленої системи може бути набір програмних засобів, що являє собою сукупність взаємозалежних процесів, що виконуються на тому самому обчислювальному пристрої. Таке уявлення дуже зручне дослідження властивостей розподілених систем.

Проте здебільшого у складі розподілених систем є кілька віддалених вузлів, пов'язаних друг з одним засобами комунікації. Незалежність або автономність вузлів є важливою особливістю розподілених систем, і ця особливість дозволяє

їм легко масштабуватись. Об'єднання системи зазвичай досягається за рахунок введення проміжного або сполучного програмного шару, в англомовній літературі відомого як middleware, який додає до загальної концепції, що лежить в основі мережі, що забезпечує однорідний вид системи. Проміжний шар включає елементи комунікаційної підсистеми та, будучи розподіленим між вузлами, маскує гетерогенність окремих вузлів, створюючи платформу для виконання розподілених додатків.

ПЕРЕВАГИ ТА НЕДОЛІКИ РC

Переваги:

- Легко масштабуються.

- Існуючі рішення інтегруються.

- Самостійне управління компонентом.

- Автономні складові частини РС.

-Гнучка й легко адаптована система.

-Поступове розширення.

Недоліки:

-Складне програмне забезпечення.

-Слабкий захист даних.

-Збій при підключенні нових компонентів.

-Складність доступу до даних.

Характеристика розподілених інформаційних систем.

Перша з таких характеристик полягає в тому, що від користувачів приховані відмінності між комп'ютерами і способи зв'язку між ними. Те ж саме відноситься і до зовнішньої організації розподілених систем. Іншою важливою характеристикою розподілених систем є спосіб, за допомогою якого користувачі і додатки одноманітно працюють в розподілених системах, незалежно від того, де і коли відбувається їх взаємодія.  
Розподілені системи повинні також відносно легко піддаватися розширенню, або масштабуванню. Ця характеристика є прямим наслідком наявності незалежних комп'ютерів, але в той же час не указує, яким чином ці комп'ютери насправді об'єднуються в єдину систему. Розподілені системи зазвичай існують постійно, проте деякі їх частини можуть тимчасово виходити з ладу. Користувачі і додатки не повинні повідомлятися про те, що ці частини замінені або полагоджені або що додані нові частини для підтримки додаткових користувачів або додатків.  
Для того, щоб підтримати представлення різних комп'ютерів і мереж у вигляді єдиної системи, організація розподілених систем часто включає додатковий рівень програмного забезпечення, що знаходиться між верхнім рівнем, на якому знаходяться користувачі і додатки, і нижнім рівнем, що складається з операційних систем

Розподілені системи мають такі характерні риси:

-просторова розподіленість компонент розподіленої системи, взасмодіють або локально, або віддалено

-компоненти розподіленої системи можуть працювати паралельно, через що швидкість роботи зростає порівняно з послідовною роботою

-кожний стан компонента розглядається локально, тобто з погляду певного обчислювального процесу, запущеного з локального робочого місця.

-компоненти працюють окремо й можуть випадати», не руйнуючи сис теми в цілому, незалежно одна від одноï, таким чином розподiленi системи підлягають частковому системному «випаданню»;

-система працює асинхронно, процеси комунікації й обробки не керуються глобальним системним часом, змiннi й процеси синхронізуються; у розподіленій системі функції керування розподіляються між різними автономними компонентами, оскільки жодна окрема компонента не може здійснювати весь контроль, що гарантує певний рівень автономії;

-розподілена система може утворюватися як об'єднання наявних систем, тобто, потрібно контекстно повне керування іменами, що дає можливість однозначно інтерпретувати найменування в рамках адміністративноï або технологiчноï області-такий випадок вважають федеративним керуванням іменами;

-для підвищення потужності розподіленої системи, програми й данi можна переміщувати мiж рiзними вузлами, таку концепцію називають міграцiсю, для підтримки якої використовують додаткові механізми, які протоколюють стан та розташування програм даних;

- розподілена система має використовувати динамічні зміни структури, ця динамічна реконфігурація потрібна, наприклад, тоді, коли протягом певного часу очікуються нові з'єднання;

- архітектура комп'ютерiв може використовувати рiзнi топології й механізми, зокрема якщо апаратура надходить від різних виробників, то що особливість називають гетерогенністю;

- розподіленій системі властива еволюцiя, вона може змінюватися протягом часу й життя;

- джерела відомостей, одиниці обробки й користувачі можуть бути фізично мобільні, а програми й дані можуть переміщуватися між вузлами для одержання даних та ресурсів з інших вузлів системи або підвищення потужностi.

Для досягнення цих характерних рис слід виконувати певні вимоги, які висуваються до розподілених систем: прозорість, відкритість, гнучкість, масштабованість, стійкість.

Приклади

Розподілені системи широко застосовують у різних сферах діяльності.

• Фінанси та комерція.

Зростання електронної комерції, наприклад такий як Amazon або eBay, і пов'язані з ними такі платіжні технології, як PayPal; пов'язане виникнення банківських та торгових технологій, а також складне поширення інформації для фінансових ринків.

• Інформаційні спільноти.

Зростання WWW як репозиторія інформації; розвиток пошукових Web-машин, таких як Google та Yandex, для пошуку в цьому великому репозиторії; виникнення цифрових бібліотек та широкомасштабне перетворення на цифровий вигляд таких традиційних джерел інформації як книги (наприклад, Google Books); значення контенту, згенерованого користувачами через сайти, такі як YouTube, Wikipedia і Flickr; виникнення соціальних мереж на основі таких сервісів, як Facebook або MySpace.

• Креативна індустрія та заходи.

Виникнення онлайн-ігор та високо інтерактивних форм заходів, доступність музики та фільмів в дома за допомогою мережевих медіацентрів або через завантаження потокового контенту; роль контенту, згенерованого користувачем (як згадувалося вище), як нова форма креативності, наприклад, через сервіси, такі як YouTube; створення нових форм мистецтва та заходів, можливих завдяки виникненню нових (включаючи мережеві) технологій.

• Охорона здоров'я.

Зростання обсягу інформації про здоров'я, ведення онлайн електронних записів пацієнтів та пов'язані з цим проблеми приватності; збільшення ролі телемедицини у підтримці віддаленої діагностики та більш просунуті послуги, такі як віддалена хірургія (включаючи спільну роботу команд охорони здоров'я); збільшення впливу додатків мережевих та вбудованих системних технологій в асистуванні життя, наприклад, моніторинг людей похилого віку в їхньому будинку.

•Утворення. Виникнення електронного навчання через, наприклад, інструменти, що базуються на Web, такі як віртуальне навчальне оточення; асоційована підтримка дистанційного навчання; підтримка спільної діяльності або навчання, що базується на створенні спільноти.

• Транспорт та логістика. Використання технологій локалізації, таких як GPS, у маршрутизації та більш загальному управлінні трафіком; системи керування сучасними транспортними засобами; розвиток сервісів карт, заснованих на Web, таких як MapQuest, Google Maps та Яндекс-карти.

• Наука. Виникнення GRID як фундаментальної технології підтримки сучасної науки, включаючи використання складних мереж комп'ютерів для зберігання, аналізу та обробки наукових даних та організації спільної роботи між групами вчених.

• Взаємодія із навколишнім середовищем. використання сенсорних технологій для моніторингу природного середовища; організація ранніх попереджень про природні лиха, таких як землетруси, затоплення, цунамі; аналіз параметрів глобального оточення для кращого розуміння складних природних явищ, як-от зміна клімату.

Як останній приклад розглянемо World Wide Web. Web надає просту, цілісну і одноманітну модель розподілених документів. Щоб побачити документ, користувачеві досить активізувати посилання. Після цього документ з'являється на екрані. У теорії (але безумовно не в поточній практиці) немає необхідності знати, з якого сервера доставляється документ, достатньо лише інформації про те, де він розташований. Публікація документа дуже проста: ви повинні тільки задати йому унікальне ім'я у формі уніфікованого покажчика ресурсу (Uniform Resource Locator, URL), яке посилається на локальний файл з вмістом документа. Якби Усесвітня павутина представлялася своїм користувачам гігантською централізованою системою документообігу, вона також могла б вважатися розподіленою системою. На жаль, цей момент ще не наступив. Так, користувачі усвідомлюють, що документи знаходяться в різних місцях і розподілені по різних серверах.

ЗАВДАННЯ РОЗПОДІЛЕНОЇ СМ

-З’єднання користувачів з ресурсами;

-Прозорість;

-Гнучкість;

-Відкритість;

-Масштабування;

Прозорість

Прозорість - полегшує взаємодію розподілених прикладних програм і приховує той факт, що процеси й ресурси фізично розподілені серед багатьох комп'ютерів.

Цей принцип дозволяє сховати складність реалізації розподілених систем від користувача.Внаслідок цього, розподілена система для користувача стає більше зручною.

Pозподілені системи, які представляються i користувачам і прикладним програмам у вигляді єдиної комп'ютерної системи, називають прозорими (transparent).

Приховування від користувачів та додатків деталей розбиття системи на компоненти та розподіл ресурсів серед безлічі комп'ютерів відоме як властивість прозорості.

Розрізняють багато характерних рис прозоростi розподілу, серед яких відокремимо такі:

-прозорість доступу приховус спецiальнi механізми доступу для локальних або віддалених служб до ресурсів системи

-прозорість мiсця розташування приховус вiд користувача системні то пологiї, тобто приховує фізичне місце розташування ресурсів (досягасться присвоєнням логiчних iмен, дозволяє замовчувати рiзнi формати даних, якi передаються рiзним процесам);

-прозорість міграції приховус факт переміщення ресурсів у інше місце;

-прозорість виконання приховус спосіб обробки обчислювальних потоків, коли обчислення виконуються паралельно або послідовно; прозорість змiни місця розташування приховус зміну ресурсу під час його обробки:

-прозорість реплікації приховує факт реплікації під час обробки даних; прозорість паралельного доступу приховує факт спільного використан ня ресурсів декількома конкуруючими процесами користувачів:

- прозорість відмови приховує факт виникнення відмов у системі й відновлення ресурсу після їх локалізації; збереження приховує факт розташування ресурсу на носіях або кешування ресурсу:

-прозорість з'єднання приховує межі між адміністративними технологічними областями в системі: прозорість групи приховує користувачів від групи під час колективнот роботи iз системою.

Гнучкість

Гнучкість– простота конфігурування системи, яка складається з різних компонентів різних виробників; можливість перенесення системи з одних операційних систем на інші, навіть в іншу файлову систему. Не має виникати ускладнень під час введення в систему нових компонентів або заміни наявних, при цьому інші компоненти, з якими не проводилося жодних дій, залишаються незмінними. Для забезпечення масштабованості слід використовувати децентралізовані дані, алгоритми, служби. У разі географічної масштабованості збільшується ризик атакування ресурсів з боку територіально віддалених користувачів, а процедура узгодження вимагає реєстрації користувачів у одному інформаційному просторі власної області або домену, а також реєстрації користувачів одного домену в іншому з наданням їм від- повідних прав доступу. Для забезпечення гнучкості розподіленої системи її слід будувати за принципом модульності, коли кожний її компонент – це певний модуль, що має своє функціональне призначення. У той же час кожний компонент підтримує принцип федеративності для співіснування автономних модулів на підставі певних стандартизованих правил. Така побудова розподілених систем забезпечує ефективну керованість, надійність функціонування системи й підтримку значень показників функціонування на заданому рівні, зокрема швидкодії.

Відкритість

Відкрита розподілена система це система, - яка пропонує стандартні засоби й служби доступу до системи широкому колу користувачів, Що використовують стандартні семантику всіх протоколів взаємодії. синтаксис

Всі протоколи взаємодії компонент усередині розподіленої системи в ідеальному випадку повинні бути засновані на загальнодоступних стандартах.

Це дозволяе використовувати для створення компонентів різні засоби розробки й операційні системи.

При порушенні цієї вимоги може стати неможливим СТВОРЕННЯ розподіленої системи, що охоплює кілька незалежних організацій.

Вважають, що розподілені системи мають бути відкритими. Відкритість комп'ютерних систем - особливість, що дозволяє легко вносити в них різні поліпшення та модифікації, забезпечувати доступність нових послуг різним клієнтським програмам. Для досягнення відкритості важливо зробити специфікації ключових інтерфейсів компонентів системи доступними для розробників програмного забезпечення.

Масштабування

Масштабування - можливість додавання в розподілену систему нових комп'ютерів для збільшення продуктивності системи, що пов'язане з поняттям балансування навантаження (load balancing) на сервери системи. До масштабування також відносяться питання ефективного розподілу ресурсів серверів, що обслуговують запити клієнтів.

Система може бути масштабованою :

-стосовно її розміру (підключення додаткових користувачів) географічно (користувачі й ресурси рознесені в просторі)

-в адміністративному змісті (проста в управлінні при роботі в ряді адміністративно незалежних

Масштабованість Розподілені системи проектуються, щоб об'єднувати велику кількість комп'ютерів. Побудова великомасштабних систем потребує правильної методології, тому надання розподіленої системі якості масштабованості є одним з найважливіших завдань при її проектуванні. Система, що масштабується, повинна гнучко адаптуватися до додавання ресурсів. Комунікаційна підсистема повинна справлятися зі збільшенням числа вузлів системи, а продуктивність розподіленої системи має збільшуватися пропорційно до їх кількості. Масштабування за розміром передбачає легкість підключення додаткових користувачів та ресурсів. Масштабована система має бути керованою, давати можливість оновлення та модифікації.

Проблеми масштабування

Питання практичної реалізації масштабованості повинні розглядатися разом з іншими вимогами, такими як безпека та продуктивність.

|  |  |
| --- | --- |
| Концепція | Приклад |
| Централізовані служби | Один сервер на всіх користувачів. Необхідно у разі збереження конфіденційних даних. |
| Централізовані дані | Єдиний телефонний довідник, доступний у режимі підключення. Централізація (наявність єдиної бази даних) у разі викличе перевантаження вхідних і вихідних ліній зв'язку. |
| Централізовані алгоритми | Організація маршрутизації на основі повної інформації про мережу. У великих системах велике число повідомлень необхідно надсилати по безлічі каналів. У таких випадках краще використовувати децентралізовані алгоритми. |
|  |

Способи масштабування

Розрізняють два способи масштабування: вертикальне та горизонтальне. Вертикальне масштабування - збільшення продуктивності кожного компонента системи з підвищення загальної продуктивності. Наприклад, для великих наборів даних це додавання великої кількості жорстких дисків до файлових серверів системи. Це найпростіший спосіб масштабування, тут не потрібні жодні зміни в прикладних програмах, проте система швидко досягає своєї межі при нарощуванні потужності вузлів.

Горизонтальне масштабування має на увазі зростання продуктивності програми за рахунок збільшення кількості вузлів. Цей спосіб масштабування може вимагати внесення змін до програм, розбиття служб на сегменти та модулі, щоб програми змогли скористатися додатковою кількістю ресурсів. Дизайн може вимагати розподілу та реплікації даних, а також наявності служб індексування та іменування для зв'язку імен ресурсів та їх локалізації. Таким чином, горизонтальне масштабування має бути внутрішнім принципом розробки системи.

Масштабована система повинна бути децентралізованою: вузли приймають рішення на основі локальної інформації і не мають повної інформації про стан системи. Централізовані компоненти, такі як сервери, служби, алгоритми, можуть стати вузьким місцем системи. Іншим важливим рішенням, особливо за наявності географічно віддалених вузлів та ненадійних з'єднань, є використання асинхронного зв'язку, коли клієнт, не будучи блокованим в очікуванні відповіді, може зайнятися альтернативною роботою. Крім того, великі сервери, наприклад веб-сервери, зазвичай мають межу кількості одночасних з'єднань, які вони можуть обслужити, і при високому трафіку ця межа може бути перевищена. Асинхронність операцій, особливо операцій читання, дозволяє серверу перемикатися між операціями швидше та обговорювати більше запитів.

Технології масштабування

Оскільки проблеми масштабованості в розподілених системах, такі як проблеми продуктивності, викликаються обмеженою потужністю серверів та мереж,

Існують три основні технології масштабування:

-приховування часу очікування зв'язку,

-розподіл

-реплікація.

**Приховування часу очікування зв'язку** застосовується у разі географічного масштабування. Основна ідея – спробувати уникнути очікування відповіді запит від віддаленого сервера. Це означає розробку програми-клієнта з використанням асинхронного зв'язку. При отриманні відповіді, програма-клієнт перерве свою роботу та викличе спеціальний обробник для завершення відправленого раніше запиту. Асинхронний зв'язок використовується в системах пакетної обробки та паралельних додатках, в яких під час очікування одним завданням завершення зв'язку передбачається виконання інших незалежних завдань. Багато завдань програми не можуть ефективно використовувати асинхронний зв'язок. Наприклад, задачі інтерактивної обробки з використанням форм. У цих випадках для скорочення обсягу взаємодії частина обчислень, що зазвичай виконуються на стороні сервера, зазвичай переміщують на бік клієнта. Особливо це стосується перевірки даних формою, тобто. ефективніше перемістити на бік клієнта перевірку даних усіх полів форми. Перенесення коду на бік клієнта зараз підтримується в Web за допомогою включення коду програми на JavaScript. Асинхронна взаємодія клієнта та сервера у Web-додатках реалізується в даний час технологією AJAX.

**Розподіл**

Передбачає розбиття компонентів більш дрібні частини з наступним їх розподілом у системі. Приклад – система доменів Інтернету (DNS). Простір DNS організований ієрархічно у вигляді дерева доменів (domains) розбитих на зони (по країнах та сферах діяльності). Імена кожної зони обробляються окремим сервером імен. Виходить, що служба доменних імен розподілена по декількох серверах, що дозволяє уникнути обробки всіх запитів одним сервером. Другий приклад – Web. Кожен документ має унікальну назву URL. Фізично середовище Web рознесене на багатьох серверах. Ім'я сервера, що містить конкретний документ, визначається за його URL-адресою. Це дозволяє збільшувати кількість документів без втрати продуктивності.

Наступна важлива технологія, використовувана підвищення продуктивності — **реплікація (копіювання)** компонентів розподіленої системи, перенесення копій даних ближче до центрів їх обробки. У сильно розосереджених системах наявність копії, що близько лежить, практично завжди веде до підвищення продуктивності. Недолік наявності кількох копій у тому, що у однієї з копій має бути поширене інші (проблема несуперечності реплік, data consistency). Проте реплікування поширене. Приватним випадком реплікації є кешування, коли репліка створюється з ініціативи клієнта — споживача ресурсу, що реплікується. Реплікація не тільки підвищує доступність, але й допомагає вирівняти завантаження компонентів, переміщуючи дані з того місця, де вони потрібні рідше, в місця, де вони потрібні частіше. Реплікація також дозволяє підвищити надійність системи та справлятися як зі збоями окремих вузлів, так і з втратою цілого кластера. Можливість паралельного читання реплікованих даних дозволяє масштабувати операції читання, а за певних умов і операції запису. Одним із технічних прийомів для підвищення продуктивності розподіленої системи є сегментування даних або шардинг - підхід, при якому кожен вузол системи містить свою частину даних та виконує операції над ними. Тут необхідно правильно розділити компоненти на частини та розподілити їх по системі. Цей метод є поширеним засобом забезпечення горизонтальної масштабованості, проте операції над кількома об'єктами можуть залучити в роботу кілька вузлів, що вимагає активної передачі даних по мережі. Крім того, при збільшенні кількості вузлів, що зберігають дані, збільшується ймовірність збоїв, що потребує надмірності - застосування реплікації. Застосування шардингу призводить до появи великої групи специфічних завдань, пов'язаних із його забезпеченням: балансування навантаження, розподіл даних та ролей, оптимізація мережного трафіку. Шардинг не гарантує знаходження шуканого блоку на вибраному вузлі, тому операції читання звертаються до деякого виділеного вузла, а той вже запитує дані або перенаправляє запит на потрібну адресу. Це уповільнює читання. Зате немає необхідності синхронізувати дані між вузлами. Крім того, частини/блоки даних можуть паралельно записуватися до різних вузлів. Це кардинально пришвидшує запис.

Припустімо, система використовує набір серверів, які обробляють запити, й одного диспетчера запитів, який розподіляє запити користувачів між серверами. Таку систему можна вважати добре масштабованою, однак диспетчер є вразливим місцем цієї системи. Натомість система з єдиним сервером може бути стійкою, якщо наявний механізм його автоматичної заміни у разі виходу з ладу.

Виконання вимог стійкості й масштабованості зазвичай спричиняє деякі додаткові витрати, що на практиці не завжди доцільно. Однак технології, які використовують під час побудови розподілених систем, мають допускати принципову можливість створення стійких і високо масштабованих систем.

Висновки

о Розподілені системи складаються з автономних комп'ютерів, які працюють спільно, створюючи уявлення про роботу у єдинiй зв'язній системі. Їхня перевага полягає в тому, що вони спрощують інтеграцію різних прикладних програм, що працюють на різних комп'ютерах, у єдину систему, добре масштабуються при їхньому правильному проектуванні.

о Розмiр розподілених систем обмежується тільки розміром базової мережі.

о Поряд зі згаданими перевагами, реалізація розподілених систем характеризується складністю програмного забезпечення, падінням продуктивності й наявністю проблем з безпекою.

о Розподілені операційні системи використовуються для управління апаратним забезпеченням взаємозалежних комп'ютерних систем, до яких відносяться мультипроцесорні й гомогенні мультикомп'ютерні системи.

о Мережні операційні системи, з іншого боку, ефективно поєднують різні комп'ютери, що працюють під управлінням своїх операційних систем, забезпечуючи користувачам доступ до локальних служб кожного з вузлів.

o Однак мережні операційні системи не створюють відчуття роботи з єдиною системою, що характерно для розподілених операційних систем.

Список використаних джерел

1. Бернс Б. Розподілені системи. Патерни проектування / Б. Бернс. – Київ: «Діалектика»
2. Коньков К.А. Розподілені системи / К.А. Коньков. М.: Вид-во «Юрайт», 2016. –
3. <http://cs.tnpu.edu.ua/index.php/%D0%A0%D0%BE%D0%B7%D0%BF%D0%BE%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D1%96_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8>
4. <https://ppt-online.org/127977>
5. <http://elartu.tntu.edu.ua/bitstream/123456789/16930/5/Mykytyshyn_A_G_Mytnyk_M_M_Kompjuterni_merezhi_Knyga_1.pdf>
6. https://studfile.net/preview/6211034/page:8/