Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка розподіленої компютерної мережі підприємства легкої промисловості»

Студентки 4 курсу, 10 групи,
спеціальності
122 «Комп'ютерні науки» _____ *підпис студента* Ужва Альона Леонідівна

Науковий керівник
кандидат педагогічних наук, доцент _____ *підпис керівника* Дивак Володимир Валерійович

Гарант освітньої програми
кандидат технічних наук, доцент _____ *підпис керівника* Демідов Павло Георгійович

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

Спеціальність 122 «Комп'ютерні науки»

Затверджую

Зав. кафедри _____Пурський О.І.

« » 202 р.

Завдання

на випускну кваліфікаційну роботу (проект) студентці

Ужва Альона Леонідівна

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

«Розробка розподіленої компютерної мережі підприємства легкої промисловості»

Затверджена наказом ректора від *«15 грудня» 2020р. № 3780*

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробити розподілену комп'ютерну мережу підприємства легкої промисловості мовою програмування Python.

Об'єкт дослідження: розробка розподіленої комп'ютерної мережі підприємств

легкої промисловості.

Предмет дослідження: методи і технологія розробки розподіленої комп'ютерної мережі підприємств легкої промисловості.

4. Перелік графічного матеріалу_____

_____

_____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Дивак В.В.		
2	Дивак В.В.		
3	Дивак В.В.		

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ЗМІСТ

Вступ

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ФОРМУВАННЯ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМСТВА

1.1. Поняття та зміст розподіленої комп'ютерної мережі підприємства

1.2. Особливості розробки комп'ютерної мережі підприємства

РОЗДІЛ 2. МОДЕЛЬ РОЗРОБКИ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМТСВА ЛЕГКОЇ ПРОМИСЛОВОСТІ

2.1. Модель розподіленої комп'ютерної мережі підприємства легкої промисловості

2.2. Методи і технологія розробки розподіленої комп'ютерної мережі підприємства легкої промисловості засобами програмування.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРОЇ МЕРЕЖІ ПІДПРИЄМСТВА ЛЕГКОЇ ПРОМИСЛОВОСТІ

3.1. Вибір і обґрунтування засобів розв'язання задачі

3.1.1. Мова програмування Python

3.1.2. Фреймворк Django

3.2. Перевірка розподіленої комп'ютерної мережі підприємства легкої промисловості мовою програмування Python

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи

		За планом	фактично
1	2	3	4
1	*Вибір теми випускної кваліфікаційної роботи*		
2	*Розробка та затвердження завдання на випускну кваліфікаційну роботу*		
3	*Вступ*		
4	*1. Аналітична частина*		
5	*2. Проектна частина*		
6	*3. Розробка*		
7	*Висновки*		
8	*Здача випускної кваліфікаційної роботи на кафедру науковому керівнику*		
9	*Попередній захист випускної кваліфікаційної роботи*		
11	*Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи*		
12	*Представлення готової зшитої випускної кваліфікаційної роботи на кафедру*		
13	*Публічний захист випускної кваліфікаційної роботи*		

8. Дата видачі завдання « » 2020 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Дивак В.В.
(прізвище, ініціали, підпис)

10. Гарант освітньої програми Демідов П.Г.
(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник Ужва А.Л.

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

_____ _____2021 р.

(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента Ужва А.Л.

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми_____Демідов П.Г.

(підпис, прізвище, ініціали)

Завідувач кафедри _____Пурський О.І.

(підпис, прізвище, ініціали)

«_____»_____2021 р.

АНОТАЦІЯ

Ужва Альона Леонідівна. «Розробка розподіленої комп'ютерної мережі підприємства легкої промисловості»

В рамках цієї дипломної роботи було розроблено модель розподіленої комп’ютерної мережі підприємства легкої промисловості, яка складається з модулів користувача і адміністратора, кожен з яких має свій функціонал, який в сукупності дає повноцінну систему обробки внутрішніх задач підприємства.

Як результат було був розроблений та створений сайт для роботи з мережею, до сайту підключена база даних, в якій зберігаються всі дані, база даних зберігається в файлах проекту на локальному сервері, що дозволить легко перенести її при побудові на хості.

КЛЮЧОВІ СЛОВА: МОВА ПРОГРАМУВАННЯ PYTHON, ФРЕЙМВОРК DJANGO, МОДЕЛЬ РОЗПОДІЛЕНОЇ КОМП’ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМСТВА ЛЕГКОЇ ПРОМИСЛОВОСТІ, ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗПОДІЛЕНОЇ КОМП’ЮТЕРОЇ МЕРЕЖІ ПІДПРИЄМСТВА ЛЕГКОЇ ПРОМИСЛОВОСТІ

ЗМІСТ

ВСТУП

Актуальність теми дослідження надзвичайно швидким ростом попиту на програмні продукти різної напрямленості.

На сьогоднішній день тема створення програмного забезпечення є надзвичайно широкою і популярною, майже всі процеси людського життя сучасники намагаються автоматизувати, спростити, переклавши частину своїх обов’язків на машини.

Створення розподілених комп’ютерних мереж стає життєвою необхідністю сучасних промислових підприємств в зв’язку з великою кількістю інформації, документі, тощо, які необхідно зберігати, категоризувати, передавати між відділами і співробітниками.

Дослідження тематики вдосконалення діяльності промислових підприємств спрямоване на впровадження інноваційного підходу із забезпеченням введення в експлуатацію розподілених комп’ютерних мереж. Саме з цією тематикою і пов’язана тема даної роботи.

Мета роботи: розробити розподілену комп'ютерну мережу підприємства легкої промисловості мовою програмування Python.

Об’єкт дослідження: розробка розподіленої комп'ютерної мережі підприємств легкої промисловості.

Предмет дослідження: методи і технологія розробки розподіленої комп'ютерної мережі підприємств легкої промисловості.

Відповідно до предмету та мети дослідження визначено наступні **завдання**:

–проаналізувати теоретичні питання розробки розподіленої комп'ютерної мережі підприємств легкої промисловості;

–розробити модель розподіленої комп'ютерної мережі підприємств легкої промисловості;

–з’ясувати особливості використання програмно-технічних засобів розподілених комп'ютерних мереж підприємства легкої промисловості;

– розробити розподілену комп'ютерну мережу підприємства легкої промисловості мовою програмування Python.

Методи дослідження: теоретичний аналіз наукової літератури, порівняння, систематизація, класифікація дали змогу дослідити і узагальнити матеріали з питань розробки розподіленої комп'ютерної мережі підприємств легкої розробити хід дослідження.

Теоретичне моделювання дало змогу розробити модель розробки розподіленої комп'ютерної мережі підприємств легкої промисловості.

Для практичного вирішення поставлених задач було використано такі методи:

ЯКІ?

Практичне значення полягає у розробці та впровадженні технології розподіленої комп'ютерної мережі підприємств легкої промисловості.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ФОРМУВАННЯ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМСТВА

1.1. Поняття та зміст розподіленої комп'ютерної мережі підприємства

Розподілені обчислення - це область інформатики, яка вивчає розподілені системи. Розподілена система - це система, компоненти якої розташовані на різних мережевих комп'ютерах, які здійснюють зв'язок та координують свої дії, передаючи повідомлення один одному з будь-якої системи. [1] Компоненти взаємодіють між собою для досягнення спільної мети. Трьома важливими характеристиками розподілених систем є: одночасність компонентів, відсутність глобального годинника та незалежний збій компонентів. [1] Приклади розподілених систем варіюються від систем, заснованих на SOA, до масових багатокористувацьких онлайн-ігор до однорангових програм.

Комп'ютерна програма, яка працює в розподіленій системі, називається розподіленою програмою (а розподілене програмування - це процес написання таких програм). [2] Існує багато різних типів реалізації механізму передачі повідомлень, включаючи чистий HTTP, коннектори, подібні до RPC, та черги повідомлень. [3]

Розподілені обчислення також стосуються використання розподілених систем для вирішення обчислювальних задач. У розподілених обчисленнях проблема ділиться на безліч завдань, кожна з яких вирішується одним або кількома комп'ютерами [4], які взаємодіють між собою за допомогою передачі повідомлень. [5]

Слово, розподілене за такими термінами, як "розподілена система", "розподілене програмування" та "розподілений алгоритм", спочатку відносилося до комп'ютерних мереж, де окремі комп'ютери фізично розподілялися в межах

певної географічної області. На сьогодні терміни використовуються у набагато ширшому розумінні, навіть стосуючись автономних процесів, які працюють на одному фізичному комп’ютері та взаємодіють між собою шляхом передачі повідомлень. [5]

Хоча не існує єдиного визначення розподіленої системи [7], наступні визначальні властивості зазвичай використовуються як:

- Існує кілька автономних обчислювальних об'єктів (комп'ютери або вузли), кожен з яких має власну локальну пам'ять. [8]
- Суб’єкти спілкуються між собою, передаючи повідомлення. [9]

Розподілена система може мати спільну мету, наприклад, вирішення великої обчислювальної проблеми; [10] тоді користувач сприймає колекцію автономних процесорів як одиницю. Як варіант, кожен комп’ютер може мати власного користувача з індивідуальними потребами, а метою розподіленої системи є координація використання спільних ресурсів або надання комунікаційних послуг для користувачів [11].

Інші типові властивості розподілених систем включають наступне:

- Система повинна терпіти збої в роботі окремих комп’ютерів. [12]
- Структура системи (топологія мережі, затримка мережі, кількість комп'ютерів) не відома заздалегідь, система може складатися з різних типів комп'ютерів та мережевих зв'язків, і система може змінюватися під час виконання розподіленої програми. [13]
- Кожен комп’ютер має лише обмежений, неповний огляд системи. Кожен комп'ютер може знати лише одну частину вводу. [14]

Розподілені системи - це групи мережевих комп'ютерів, які мають спільну мету своєї роботи. Терміни "одночасні обчислення", "паралельні обчислення" та "розподілені обчислення" мають значне перекриття, і між ними не існує чітких

відмінностей. [15] Одна і та ж система може характеризуватися як "паралельна", так і "розподілена"; процесори в типовій розподіленій системі працюють паралельно. [16] Паралельні обчислення можуть розглядатися як особлива тісно пов'язана форма розподілених обчислень [17], а розподілені обчислення можуть розглядатися як слабо пов'язана форма паралельних обчислень. [7] Тим не менше, можна грубо класифікувати паралельні системи як "паралельні" або "розподілені", використовуючи такі критерії:

При паралельних обчисленнях усі процесори можуть мати доступ до спільної пам'яті для обміну інформацією між процесорами. [18]

У розподілених обчисленнях кожен процесор має власну приватну пам’ять (розподілену пам’ять). Інформація обмінюється шляхом передачі повідомлень між процесорами. [19]

Рис. 1.1 ілюструє різницю між розподіленою та паралельною системами. Фігура (a) - схематичний вигляд типової розподіленої системи; система представлена у вигляді топології мережі, в якій кожен вузол є комп’ютером, а кожна лінія, що з’єднує вузли, є каналом зв'язку. На рисунку (b) більш детально показана одна і та ж розподілена система: кожен комп’ютер має власну локальну пам’ять, і інформацією можна обмінюватися лише шляхом передачі повідомлень з одного вузла на інший за допомогою доступних каналів зв'язку. На рисунку (в) показана паралельна система, в якій кожен процесор має прямий доступ до спільної пам'яті.

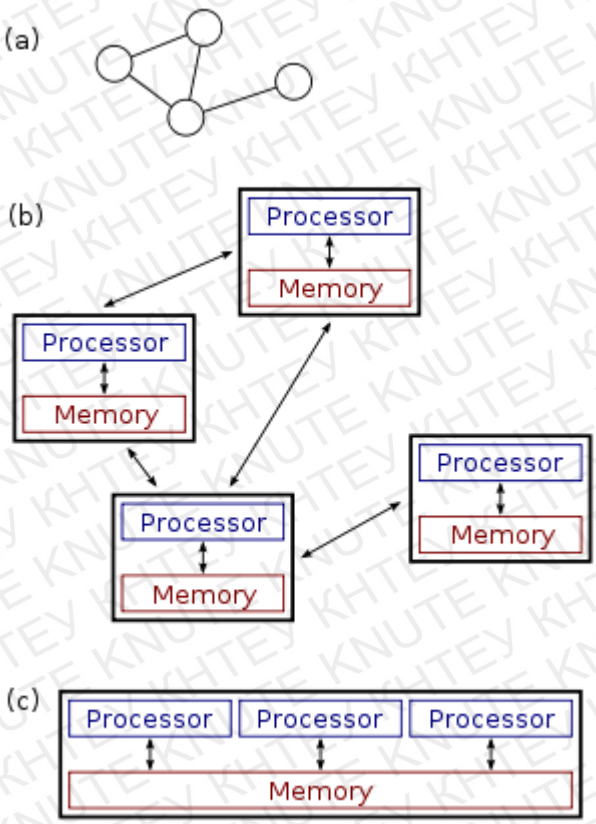

Рис. 1.1 — Розподілені системи

Ситуація ускладняється ще й традиційним використанням термінів паралельний та розподілений алгоритм, який не зовсім відповідає наведеним вище визначенням паралельних та розподілених систем (див. Нижче для більш детального обговорення). Тим не менше, як правило, високопродуктивні паралельні обчислення в мультипроцесорі із спільною пам'яттю використовують паралельні алгоритми, тоді як координація широкомасштабної розподіленої системи використовує розподілені алгоритми. [20]

Використання одночасних процесів, які взаємодіють за допомогою передачі повідомлень, сягає своїм корінням в архітектури операційних систем, вивчені в 1960-х рр. [21] Першими розповсюдженими розподіленими системами були локальні мережі, такі як Ethernet, який був винайдений у 1970-х рр. [22]

ARPANET, один з попередників Інтернету, був представлений наприкінці 1960-х, а електронна пошта ARPANET була винайдена на початку 1970-х. Електронна пошта стала найуспішнішим додатком ARPANET [23], і це, мабуть, перший приклад широкомасштабного розподіленого додатка. На додаток до ARPANET (та його наступника, глобального Інтернету), до інших ранніх світових комп'ютерних мереж належали Usenet та FidoNet з 1980-х років, які обидва використовувались для підтримки розподілених систем обговорення.

Дослідження розподілених обчислень стало власною галуззю інформатики наприкінці 1970-х - на початку 1980-х. Перша конференція в галузі, Симпозіум з принципів розподіленого обчислення (PODC), датується 1982 роком, а аналогічний Міжнародний симпозіум з розподілених обчислень (DISC) вперше відбувся в Оттаві в 1985 році як Міжнародний семінар з розподілених алгоритмів на графіках. [25]

1.2. Особливості розробки комп’ютерної мережі підприємства

Для розподілених обчислень використовуються різні апаратні та програмні архітектури. На нижчому рівні необхідно з'єднати декілька процесорів з якоюсь мережею, незалежно від того, надрукована ця мережа на друкованій платі або складається з нещільно з'єднаних пристроїв та кабелів. На більш високому рівні необхідно пов’язати процеси, що працюють на цих центральних процесорах, з якоюсь системою зв'язку. [26]

Розподілене програмування, як правило, підпадає під одну з кількох базових архітектур: клієнт-сервер, трирівневу, n-рівневу або однорангову; або категорії: вільна муфта або герметична муфта. [27]

Клієнт-сервер: архітектури, при яких розумні клієнти звертаються до сервера для отримання даних, а потім форматують і відображають їх користувачам. Вхідні дані на клієнті передаються назад на сервер, коли це означає постійну зміну.

Трирівневі: архітектури, які переміщують клієнтський інтелект до середнього рівня, щоб можна було використовувати клієнтів без громадянства. Це спрощує розгортання додатків. Більшість веб-додатків трирівневі.

n-рівень: архітектури, які зазвичай стосуються веб-додатків, які надалі пересилають свої запити до інших корпоративних служб. Цей тип додатків є одним з найбільш відповідальних за успіх серверів додатків.

Peer-to-peer: архітектури, в яких немає спеціальних машин, що надають послугу або керують мережевими ресурсами. [28]: 227 Натомість усі обов'язки рівномірно розподілені між усіма машинами, відомими як однорангові. Ровесники можуть служити як клієнтами, так і серверами. [29] Приклади цієї архітектури включають BitTorrent та мережу біткойнів.

Іншим основним аспектом розподіленої обчислювальної архітектури є метод передачі та координації роботи між одночасними процесами. За допомогою різних протоколів передачі повідомлень процеси можуть взаємодіяти безпосередньо один з одним, як правило, у відносинах головний / підлеглий. Крім того, архітектура, орієнтована на базу даних, може дозволити розподіленим обчисленням здійснюватись без будь-якої форми прямого міжпроцесорного зв'язку, використовуючи спільну базу даних. [30] Зосереджена на базі даних архітектура, зокрема, забезпечує аналіз реляційної обробки в схематичній архітектурі, що забезпечує ретрансляцію середовища. Це дозволяє розподіленим обчислювальним функціям як усередині параметрів мережевої бази даних, так і поза ними. [31]

Причини використання розподілених систем та розподілених обчислень можуть включати:

Сама природа програми може вимагати використання комунікаційної мережі, яка з'єднує кілька комп'ютерів: наприклад, дані, отримані в одному фізичному місці та необхідні в іншому місці.

Є багато випадків, коли використання одного комп’ютера було б принципово можливим, але використання розподіленої системи корисно з практичних міркувань. Наприклад, може бути вигідніше отримати бажаний рівень продуктивності за допомогою кластера декількох комп'ютерів низького класу в порівнянні з одним комп'ютером високого класу. Розподілена система може забезпечити більшу надійність, ніж нерозподілена система, оскільки немає жодної точки відмови. Більше того, розподілену систему може бути простіше розширити та керувати нею, ніж монолітну однопроцесорну систему. [32]

РОЗДІЛ 2. МОДЕЛЬ РОЗРОБКИ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМТСВА ЛЕГКОЇ ПРОМИСЛОВОСТІ

2.1. Модель розподіленої комп'ютерної мережі підприємства легкої промисловості

Розподілена комп'ютерна мережа підприємства легкої промисловості повинна мати наступний функціонал:

- зберігання в базі знань інформації про користувачів системи.;
- надання рядовому користувачеві доступу до активних промислових задач і подання відгуків по ним;
- надання адміністратору можливості створювати нові задачі, переглядати та редагувати старі.

Існує також ряд додаткових вимог, пов'язаних з особливостями його використання в web-системі. Додаток буде використовуватися великим числом користувачів, проте всі вони повинні будуть мати доступ до загальної інформації. При цьому потрібно, щоб користувачі мали можливість додавати дані і ці зміни були видні адміністратору. Таким чином, необхідно забезпечити централізоване зберігання даних, організувати можливість одночасного доступу до них декількох користувачів, а також запобігти виникненню конфліктних ситуацій.

На основі аналізу загальних вимог до моделювання інтерфейсної частини web-системи, було створено загальний та простий вигляд, без зайвих деталей в інтерфейсі.

Необхідність поновлення сторінки порушує нормальний хід думки. Тому є сенс заповнювати сторінки динамічним контентом, щоб позбутися зайвих її перезавантажень.

Нижче наведені вимоги до інтерфейсу саме того додатку, що розробляється:

- наявність форми авторизації;
- наявність форми реєстрації;
- наявність сторінки користувача зі список задач;
- наявність форми відповіді на задачу;
- наявність сторінки адміністратора.

Звичайно, всі форми для створення і редагування повинні містити необхідну довідкову інформацію та підказки для спрощення роботи з ними.

Сучасні інформаційні технології дають змогу створювати єдине інформаційне середовище в закладах середньої освіти (фізичну основу яких становлять інтегровані комп'ютерні мережі та системи зв'язку), яке допомагає в динаміці супроводжувати та координувати як внутрішню, так і зовнішню діяльність. Зокрема, цей підхід включає технічну, організаційну та методологічну інтеграцію таких базових напрямів управлінської діяльності, як виробнича, організаційна, маркетингова, фінансова, бухгалтерська, кадрова. Інформаційні ресурси розміщуються в розподілених базах даних, які працюють у полі єдиних протоколів і правил під керівництвом адміністратора мережі.

Завдяки різноманітним архітектурним рішенням, функціональні навантаження системи гармонійно розподіляються між модулями та структурними елементами.

Прикладом такого архітектурного рішення є архітектура «клієнт-сервер».

Клієнт-серверна модель - це розподілена структура додатків, яка розділяє завдання або робочі навантаження між провайдерами ресурсу чи послуги, які називаються серверами, та запитувачами послуг, які називаються клієнтами. Часто клієнти та сервери спілкуються через комп'ютерну мережу на окремому

обладнанні, але і клієнт, і сервер можуть перебувати в одній системі. Хост сервера запускає одну або кілька серверних програм, які діляться своїми ресурсами з клієнтами. Клієнт, як правило, не ділиться жодним із своїх ресурсів, але він запитує вміст або послугу у сервера. Отже, клієнти ініціюють сеанси зв'язку із серверами, які очікують на вхідні запити. Прикладами комп'ютерних програм, що використовують модель клієнт-сервер, є електронна пошта, мережевий друк та Всесвітня павутина.

Характеристика "клієнт-сервер" описує взаємозв'язок взаємодіючих програм у програмі. Серверний компонент надає функцію або послугу одному або багатьом клієнтам, які ініціюють запити на такі послуги. Сервери класифікуються за послугами, які вони надають. Наприклад, веб-сервер обслуговує веб-сторінки, а файловий - комп'ютерні. Спільним ресурсом може бути будь-яке програмне забезпечення серверного комп'ютера та електронні компоненти, починаючи від програм та даних, закінчуючи процесорами та запам'ятовуючими пристроями. Спільне використання ресурсів сервера є послугою.

Чи комп’ютер є клієнтом, сервером чи обома, визначається характером програми, яка вимагає сервісних функцій. Наприклад, на одному комп’ютері може одночасно працювати веб-сервер та програмне забезпечення файлового сервера, щоб обслуговувати різні дані клієнтам, які роблять різні типи запитів. Клієнтське програмне забезпечення також може взаємодіяти із серверним програмним забезпеченням на тому самому комп'ютері. Зв'язок між серверами, наприклад для синхронізації даних, іноді називають міжсерверним або міжсерверним.

Загалом, послуга - це абстракція комп'ютерних ресурсів, і клієнт не повинен займатися тим, як працює сервер під час виконання запиту та надання

відповіді. Клієнт повинен розуміти відповідь лише на основі відомого протоколу програми, тобто вмісту та форматування даних для запитуваної послуги.

Клієнти та сервери обмінюються повідомленнями за шаблоном обміну повідомленнями запит-відповідь. Клієнт надсилає запит, а сервер повертає відповідь. Цей обмін повідомленнями є прикладом міжпроцесорного спілкування. Для спілкування комп’ютери повинні мати спільну мову та дотримуватися правил, щоб і клієнт, і сервер знали, чого очікувати. Мова та правила спілкування визначені протоколом зв'язку. Усі протоколи клієнт-сервер працюють на рівні програми. Протокол рівня додатків визначає основні схеми діалогу. Щоб ще більше формалізувати обмін даними, сервер може реалізувати інтерфейс прикладного програмування (API). API - це рівень абстракції для доступу до послуги. Обмежуючи спілкування певним форматом вмісту, це полегшує розбір. Абстрагуючи доступ, це полегшує міжплатформенний обмін даними.

Сервер може отримувати запити від багатьох різних клієнтів за короткий проміжок часу. Комп’ютер може виконувати обмежену кількість завдань у будь-який момент і покладається на систему планування, щоб визначити пріоритет вхідних запитів клієнтів для їх задоволення. Щоб запобігти зловживанням та максимізувати доступність, серверне програмне забезпечення може обмежити доступність для клієнтів. Атаки відмови в службі призначені для використання зобов'язання сервера обробляти запити, перевантажуючи його надмірними частотами запитів. Шифрування слід застосовувати, якщо конфіденційна інформація повинна передаватися між клієнтом та сервером.

Коли клієнт банку отримує доступ до послуг онлайн-банкінгу за допомогою веб-браузера (клієнта), клієнт ініціює запит на веб-сервер банку. Реєстраційні дані клієнта можуть зберігатися в базі даних, і веб-сервер отримує

доступ до сервера баз даних як клієнт. Сервер додатків інтерпретує повернені дані, застосовуючи бізнес-логіку банку, і забезпечує вихід веб-сервера. Нарешті, веб-сервер повертає результат для відображення клієнтському веб-браузеру.

На кожному кроці цієї послідовності обміну повідомленнями клієнт-сервер комп'ютер обробляє запит і повертає дані. Це шаблон обміну повідомленнями запит-відповідь. Коли всі запити задоволені, послідовність завершена, і веб-браузер представляє дані клієнту.

Цей приклад ілюструє шаблон дизайну, застосовний до моделі клієнт-сервер: розділення проблем.

Модель клієнт-сервер не диктує, що сервери-хости повинні мати більше ресурсів, ніж клієнт-хости. Швидше, це дозволяє будь-якому комп’ютеру загального призначення розширити свої можливості за допомогою спільних ресурсів інших хостів. Однак централізовані обчислення виділяють велику кількість ресурсів на невелику кількість комп’ютерів. Чим більше обчислень завантажується з клієнт-хостів на центральні комп'ютери, тим простішими можуть бути клієнт-хости. Для обчислень та зберігання він значною мірою покладається на мережеві ресурси (сервери та інфраструктуру). Бездисковий вузол завантажує з мережі навіть свою операційну систему, а комп'ютерний термінал взагалі не має операційної системи; це лише інтерфейс вводу / виводу для сервера. На відміну від цього, товстий клієнт, такий як персональний комп’ютер, має багато ресурсів і не покладається на сервер для основних функцій.

У міру зниження ціни та збільшення потужності мікрокомп’ютерів з 1980-х до кінця 1990-х років багато організацій перевели обчислення від централізованих серверів, таких як мейнфрейми та міні-комп’ютери, до жирних клієнтів. Це дало більше, більш індивідуалізоване панування над комп’ютерними

ресурсами, але складне управління інформаційними технологіями. Протягом 2000-х років веб-додатки визріли настільки, щоб конкурувати з прикладним програмним забезпеченням, розробленим для конкретної мікроархітектури. Це дозрівання, більш доступне масове сховище та поява сервісно-орієнтованої архітектури були одними з факторів, що породили тенденцію хмарних обчислень 2010-х років.

На додаток до моделі клієнт-сервер, розподілені обчислювальні програми часто використовують архітектуру однорангових (P2P) додатків.

У моделі клієнт-сервер сервер часто призначений для роботи як централізована система, яка обслуговує багатьох клієнтів. Витрати на обчислювальну потужність, пам’ять та пам’ять сервера повинні бути відповідно масштабовані відповідно до очікуваного навантаження. Системи балансування навантаження та відмови часто використовуються для масштабування сервера за межі однієї фізичної машини.

Балансування навантаження визначається як методичний та ефективний розподіл мережевого або додаткового трафіку на декількох серверах у фермі серверів. Кожен балансир навантаження знаходиться між клієнтськими пристроями та серверними серверами, отримуючи та розподіляючи вхідні запити на будь-який доступний сервер, здатний їх виконати.

У одноранговій мережі два або більше комп’ютери (однорангові мережі) об’єднують свої ресурси та спілкуються в децентралізованій системі. Піри - це рівноправні або еквіпотентні вузли в неієрархічній мережі. На відміну від клієнтів у мережі клієнт-сервер або клієнт-черга – клієнт, однорангові комунікації здійснюють безпосередню взаємодію. [Необхідне цитування] У однорангових мережах алгоритм протоколу однорангових комунікацій врівноважує навантаження і навіть однолітки зі скромними ресурсами можуть

допомогти розподілити навантаження. Якщо вузол стає недоступним, його спільні ресурси залишаються доступними до тих пір, поки його пропонують інші партнери. В ідеалі, рівному рівному користувачеві не потрібно досягати високої доступності, оскільки інші зайві однорангові компенсують час простою будь-якого ресурсу; оскільки доступність та завантажуваність однолітків змінюються, протокол перенаправляє запити.

І клієнт-сервер, і ведучий-підлеглий розглядаються як підкатегорії розподілених однорангових систем.

2.2. Методи і технологія розробки розподіленої комп'ютерної мережі підприємства легкої промисловості засобами програмування.

Веб-сайт - це сукупність веб-сторінок та пов'язаного вмісту, які ідентифікуються загальним доменним ім'ям та публікуються принаймні на одному веб-сервері. Помітними прикладами є wikipedia.org, google.com та amazon.com.

Усі загальнодоступні веб-сайти в сукупності складають Всесвітню павутину. Існують також приватні веб-сайти, доступ до яких доступний лише в приватній мережі, наприклад, внутрішній веб-сайт компанії для її співробітників.

Веб-сайти, як правило, присвячені певній темі або меті, такі як новини, освіта, комерція, розваги чи соціальні мережі. Гіперпосилання між веб-сторінками спрямовує навігацію по сайту, яке часто починається з домашньої сторінки.

Користувачі можуть отримати доступ до веб-сайтів на різних пристроях, включаючи настільні, ноутбуки, планшети та смартфони. Програма, що використовується на цих пристроях, називається веб-браузером.

Всесвітня павутина (WWW) була створена в 1990 році британським фізиком ЦЕРН Тімом Бернерсом-Лі. [1] 30 квітня 1993 р. ЦЕРН оголосив, що Всесвітня павутина буде безкоштовною для використання будь-ким [2]. До впровадження протоколу передачі гіпертексту (HTTP) для отримання окремих файлів із сервера використовувались інші протоколи, такі як протокол передачі файлів та протокол gopher. Ці протоколи пропонують просту структуру каталогів, в якій користувач переходить і де він обирає файли для завантаження. Документи найчастіше подавалися у вигляді текстових файлів без форматування або кодувались у форматах текстового процесора.

Веб-сайти можуть використовуватися по-різному: персональний веб-сайт, корпоративний веб-сайт компанії, урядовий веб-сайт, веб-сайт організації тощо. Веб-сайти можуть бути роботою приватної особи, бізнесу чи іншої організації і, як правило, присвячені конкретна тема або мета. Будь-який веб-сайт може містити гіперпосилання на будь-який інший веб-сайт, тому різниця між окремими сайтами, як сприймається користувачем, може бути розмитою.

Деякі веб-сайти вимагають реєстрації користувачів або передплати для доступу до вмісту. Приклади веб-сайтів, на які здійснюється підписка, включають багато бізнес-сайтів, веб-сайти новин, веб-сайти академічних журналів, ігрові веб-сайти, веб-сайти спільного використання файлів, дошки оголошень, веб-адреси електронної пошти, веб-сайти соціальних мереж, веб-сайти, що надають дані про фондовий ринок у режимі реального часу, а також веб-сайти, що надають різні інші послуги.

Хоча "веб-сайт" був оригінальним написанням (іноді з великої літери "Веб-сайт", оскільки "Веб" є власним іменником, коли йдеться про Всесвітню павутину), цей варіант став рідкісним, а "веб-сайт" став стандартним написанням . Усі основні керівництва стилем, такі як Чиказький посібник стилю [3] та AP Stylebook [4], відображають цю зміну.

Статичний веб-сайт - це веб-сторінки, які мають веб-сторінки, що зберігаються на сервері у форматі, який надсилається клієнтському веб-браузеру. В основному він кодується мовою розмітки гіпертексту (HTML); Каскадні таблиці стилів (CSS) використовуються для контролю зовнішнього вигляду за базовим HTML. Зображення зазвичай використовуються для отримання бажаного вигляду та як частина основного змісту. Аудіо чи відео також можна вважати "статичним" вмістом, якщо він відтворюється автоматично або, як правило, є неінтерактивним. Цей тип веб-сайтів зазвичай відображає однакову

інформацію для всіх відвідувачів. Подібно до роздачі друкованої брошури клієнтам або клієнтам, статичний веб-сайт, як правило, надає послідовну стандартну інформацію протягом тривалого періоду часу. Незважаючи на те, що власник веб-сайту може періодично робити оновлення, редагування тексту, фотографій та іншого вмісту здійснюється вручну, і може знадобитися базові навички дизайну веб-сайту та програмне забезпечення. Прості форми або маркетингові приклади веб-сайтів, такі як класичний веб-сайт, веб-сайт із п'ятьма сторінками або веб-сайт брошури, часто є статичними веб-сайтами, оскільки вони представляють користувачеві заздалегідь визначену, статичну інформацію. Це може включати інформацію про компанію та її продукти та послуги через текст, фотографії, анімацію, аудіо / відео та навігаційні меню.

Статичні веб-сайти все ще можуть використовувати серверні компоненти (SSI) як зручність редагування, наприклад, спільний доступ до загального рядка меню на багатьох сторінках. Оскільки поведінка сайту до читача все ще залишається статичним, це не вважається динамічним сайтом.

Динамічний веб-сайт - це веб-сайт, який часто і автоматично змінюється або налаштовується. Динамічні сторінки на сервері генеруються "на льоту" за допомогою комп'ютерного коду, який виробляє HTML (CSS відповідає за зовнішній вигляд і, отже, статичні файли). Існує широкий спектр програмних систем, таких як CGI, Java-сервлети та Java Server Pages (JSP), Active Server Pages і ColdFusion (CFML), які доступні для створення динамічних веб-систем та динамічних веб-сайтів. Різні фреймворки веб-програм та системи веб-шаблонів доступні для загальновживаних мов програмування, таких як Perl, PHP, Python та Ruby, щоб полегшити та спростити створення складних динамічних веб-сайтів.

Сайт може відображати поточний стан діалогу між користувачами, відстежувати мінливу ситуацію або надавати інформацію певним чином персоналізовану відповідно до вимог окремого користувача. Наприклад, коли запитується головна сторінка сайту новин, код, що працює на веб-сервері, може поєднувати збережені фрагменти HTML із новинами, отриманими з бази даних або іншого веб-сайту за допомогою RSS, щоб створити сторінку, що включає найсвіжішу інформацію. Динамічні сайти можуть бути інтерактивними, використовуючи HTML-форми, зберігаючи та зчитуючи файли cookie браузера, або створюючи ряд сторінок, що відображають попередню історію кліків. Інший приклад динамічного вмісту - це коли роздрібний веб-сайт з базою даних медіа-продуктів дозволяє користувачеві вводити запит на пошук, наприклад за ключовим словом "Бітлз". У відповідь вміст веб-сторінки спонтанно змінить те, як він виглядав раніше, а потім відобразить список продуктів "Бітлз", таких як компакт-диски, DVD-диски та книги. Динамічний HTML використовує код JavaScript, щоб вказувати веб-браузеру, як інтерактивно змінювати вміст сторінки. Одним із способів моделювання певного типу динамічного веб-сайту, уникаючи втрати продуктивності ініціювання динамічного механізму для кожного користувача або підключення, є періодична автоматична регенерація великої серії статичних сторінок.

Ранні веб-сайти мали лише текст, а незабаром і зображення. Потім плагіни веб-браузера використовувались для додавання аудіо, відео та інтерактивності (наприклад, для багатофункціональної програми Інтернету, яка відображає складність настільної програми, як текстовий процесор). Прикладами таких плагінів є Microsoft Silverlight, Adobe Flash, Adobe Shockwave та аплети, написані на Java. HTML 5 містить положення щодо аудіо та відео без плагінів. JavaScript також вбудований у більшість сучасних веб-браузерів і дозволяє творцям веб-

сайтів надсилати код веб-браузеру, який вказує йому, як інтерактивно змінювати вміст сторінки та спілкуватися з веб-сервером, якщо це необхідно. Внутрішнє представлення вмісту браузера відомо як об’єктна модель документа (DOM), а техніка - динамічний HTML.

WebGL (Web Graphics Library) - це сучасний API JavaScript для візуалізації інтерактивної 3D-графіки без використання плагінів. Це дозволяє інтерактивний контент, такий як 3D-анімація, візуалізація та відео-пояснення, представленим користувачам найбільш інтуїтивно зрозумілим способом. [5]

Тенденція 2010 року на веб-сайтах під назвою "адаптивний дизайн" забезпечила найкращий досвід перегляду, оскільки надає користувачам макет на основі пристрою. Ці веб-сайти змінюють свій макет відповідно до пристрою або мобільної платформи, що забезпечує багатий досвід користувачів. [6]

Веб-сайти можна розділити на дві великі категорії - статичні та інтерактивні. Інтерактивні сайти є частиною спільноти веб-сайтів Web 2.0 і дозволяють взаємодіяти між власником сайту та відвідувачами або користувачами сайту. Статичні сайти обслуговують або збирають інформацію, але не дозволяють взаємодіяти з аудиторією або користувачами безпосередньо. Деякі веб-сайти є інформаційними або виготовляються ентузіастами або для особистого користування чи розваги. Багато веб-сайтів мають на меті заробляти гроші, використовуючи одну або кілька бізнес-моделей, зокрема:

- Розміщення цікавого контенту та продаж контекстної реклами або шляхом прямих продажів, або через рекламну мережу.
- Електронна комерція: товари чи послуги купуються безпосередньо через веб-сайт
- Реклама товарів або послуг, доступних у цегельному та будівельному бізнесі

- Freemium: базовий вміст доступний безкоштовно, але преміум-вміст вимагає оплати (наприклад, веб-сайт WordPress, це платформа з відкритим кодом для створення блогу чи веб-сайту).

Деякі веб-сайти можуть бути включені в одну або кілька з цих категорій.

Наприклад, веб-сайт бізнесу може рекламувати продукцію бізнесу, але також може розміщувати інформативні документи, такі як технічні документи. Існують також численні підкатегорії до перерахованих вище. Наприклад, порносайт - це певний тип веб-сайту електронної комерції або бізнес-сайту (тобто він намагається продати членство для доступу до свого сайту) або має можливості соціальних мереж. Фан-сайт може бути посвятою власника певній знаменитості. Веб-сайти обмежені архітектурними обмеженнями (наприклад, обчислювальна потужність, присвячена веб-сайту). Дуже великі веб-сайти, такі як Facebook, Yahoo !, Microsoft та Google, використовують багато серверів та обладнання для балансування навантажень, наприклад, комутатори служб вмісту Cisco, для розподілу навантажень відвідувачів на декілька комп’ютерів у різних місцях. На початок 2011 року Facebook використовував 9 центрів обробки даних із приблизно 63000 серверами.

У лютому 2009 року компанія Netcraft, компанія з моніторингу Інтернету, яка відслідковувала зростання Інтернету з 1995 р., Повідомила, що в 2009 р. Існувало 215 675 903 веб-сайтів з доменними іменами та вмістом на них, порівняно з лише 19 732 веб-сайтами в серпні 1995 р. [8] Після досягнення 1 мільярда веб-сайтів у вересні 2014 року, етап, підтверджений NetCraft в огляді веб-серверів за жовтень 2014 року, і те, що Інтернет-статистика стала першим, хто повідомив про це, про що свідчить цей твіт від самого винахідника Всесвітньої мережі, Тіма Бернерса Лі - кількість веб-сайтів у світі згодом зменшилась, повернувшись до рівня нижче 1 мільярда. Це пов’язано з

щомісячними коливаннями кількості неактивних веб-сайтів. Кількість веб-сайтів до березня 2016 р. Продовжувала зростати до понад 1 млрд. І з тих пір продовжує зростати [9].

Веб-розробка - це робота, пов’язана з розробкою веб-сайту для Інтернету (Всесвітня павутина) або інтрамережі (приватна мережа). [1] Веб-розробка може варіюватися від розробки простої однієї статичної сторінки простого тексту до складних Інтернет-програм (Інтернет-додатків), електронного бізнесу та послуг соціальних мереж. Більш повний перелік завдань, до яких зазвичай відноситься веб-розробка, може включати веб-інженерію, веб-дизайн, розробку веб-вмісту, зв'язок з клієнтом, сценарії на стороні клієнта / сервера, налаштування безпеки веб-сервера та мережі та розвиток електронної комерції.

Серед веб-спеціалістів "веб-розробка" зазвичай відноситься до основних недизайнерських аспектів побудови веб-сайтів: написання розмітки та кодування. [2] Веб-розробка може використовувати системи управління вмістом (CMS), щоб зробити зміст вмістом простішим та доступнішим з базовими технічними навичками.

Для великих організацій та бізнесу групи веб-розробників можуть складатися з сотень людей (веб-розробників) і дотримуватися стандартних методів, таких як Agile, під час розробки веб-сайтів. Менші організації можуть вимагати лише одного постійного або підрядного розробника, або вторинне призначення на відповідні посади, такі як графічний дизайнер або технік інформаційних систем. Веб-розробка може бути спільним зусиллям між департаментами, а не областю призначеного департаменту. Існує три різновиди спеціалізації веб-розробників: розробник з інтерфейсу, розробник з інтерфейсу та розробник із повним стеком. Інтернетні розробники несуть відповідальність за

поведінку та візуальні ефекти, які працюють у браузері користувача, тоді як інтерфейсні розробники мають справу з серверами.

З моменту комерціалізації Інтернету веб-розробка стала зростаючою галуззю. Зростання цієї галузі відбувається за рахунок підприємств, які бажають використовувати свій веб-сайт для реклами та продажу товарів та послуг споживачам. [3]

Існує багато інструментів з відкритим кодом для веб-розробки, таких як BerkeleyDB, GlassFish, стек LAMP (Linux, Apache, MySQL, PHP) та Perl / Plack. Це дозволило звести витрати на навчання веб-розробці до мінімуму. Ще одним фактором, що сприяє зростанню галузі, стало зростання простого у використанні програмного забезпечення для веб-розробки WYSIWYG, такого як Adobe Dreamweaver, BlueGriffon та Microsoft Visual Studio. Для використання такого програмного забезпечення все ще необхідні знання мови розмітки HyperText (HTML) або мов програмування, але основи можна швидко вивчити та впровадити.

Постійно зростаючий набір інструментів та технологій допоміг розробникам створювати більш динамічні та інтерактивні веб-сайти. Крім того, веб-розробники тепер допомагають надавати програми як веб-служби, які традиційно були доступні лише як програми на настільному комп’ютері. Це дало багато можливостей для децентралізації розповсюдження інформації та засобів масової інформації. Приклади можна побачити із зростанням хмарних сервісів, таких як Adobe Creative Cloud, Dropbox та Google Drive. Ці веб-служби дозволяють користувачам взаємодіяти з програмами з багатьох місць, замість того, щоб бути прив’язаними до певної робочої станції для свого середовища програм.

Прикладами кардинальних перетворень у спілкуванні та комерції на чолі з веб-розробкою є електронна комерція. Інтернет-сайти аукціонів, такі як eBay, змінили спосіб споживачів знаходити та купувати товари та послуги. Інтернет-магазини, такі як Amazon.com та Buy.com (серед багатьох інших), змінили досвід покупок та торгівлі для багатьох споживачів. Іншим прикладом трансформаційного спілкування, керованого веб-розробкою, є блог. Веб-програми, такі як WordPress та Movable Type, створили середовища для ведення блогів для окремих веб-сайтів. Поширене використання систем управління вмістом з відкритим кодом та систем управління корпоративним контентом розширило вплив веб-розробки на онлайн-взаємодію та спілкування.

Розробка веб-сайтів також вплинула на особисті мережі та маркетинг. Веб-сайти вже не є просто інструментами для роботи чи комерції, а ширше служать для спілкування та соціальних мереж. Такі веб-сайти, як Facebook та Twitter, надають користувачам платформу для спілкування, а організації - більш особистий та інтерактивний спосіб залучення громадськості.

Веб-розробка враховує багато міркувань безпеки, таких як перевірка помилок при введенні даних через форми, фільтрація вихідних даних та шифрування. Шкідливі практики, такі як введення SQL, можуть виконуватися користувачами з ненавмисними намірами, але лише з примітивними знаннями про веб-розробку в цілому. Сценарії можна використовувати для використання веб-сайтів, надаючи несанкціонований доступ зловмисним користувачам, які намагаються збирати таку інформацію, як адреси електронної пошти, паролі та захищений вміст, наприклад номери кредитних карток.

Дещо з цього залежить від серверного середовища, на якому запущена мова сценаріїв, наприклад ASP, JSP, PHP, Python, Perl або Ruby, і тому не обов'язково підтримувати самого розробника. Однак рекомендується жорстке тестування

веб-додатків перед публічним випуском, щоб запобігти подібним подвигам.

Якщо на веб-сайті є якась контактна форма, вона повинна включати в неї поле captcha, яке не дозволяє комп'ютерним програмам автоматично заповнювати форми, а також розсилати пошту.

Захист веб-сервера від вторгнень часто називають зміцненням порту сервера. Багато технологій застосовуються для захисту інформації в Інтернеті, коли вона передається з одного місця в інше. Наприклад, сертифікати TLS (або "сертифікати SSL") видаються органами сертифікації для запобігання шахрайству в Інтернеті. Багато розробників часто використовують різні форми шифрування при передачі та зберіганні конфіденційної інформації. Базове розуміння питань безпеки інформаційних технологій часто є частиною знань веб-розробника.

Оскільки нові веб-програми виявляють нові діри в безпеці навіть після тестування та запуску, оновлення виправлень безпеки часто трапляються для широко використовуваних програм. Часто робота веб-розробників - постійно оновлювати програми, коли випускаються виправлення безпеки та виявляються нові проблеми безпеки.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРОЇ МЕРЕЖІ ПІДПРИЄМСТВА ЛЕГКОЇ ПРОМИСЛОВОСТІ

3.1. Вибір і обґрунтування засобів розв'язання задачі

3.1.1. Мова програмування Python

Python - інтерпретована мова програмування високого рівня та загального призначення. Філософія дизайну Python наголошує на читабельності коду завдяки помітному використанню значних відступів. Його мовні конструкції та об'єктно-орієнтований підхід мають на меті допомогти програмістам написати чіткий логічний код для малих та великих проектів.

Python динамічно набирається та збирається сміття. Він підтримує декілька парадигм програмування, включаючи структуроване (зокрема, процедурне), об'єктно-орієнтоване та функціональне програмування. Python часто описують як мову, що включає батареї, завдяки своїй повній стандартній бібліотеці.

Гідо ван Россум почав працювати над Python наприкінці 1980-х років, як наступник мови програмування ABC, і вперше випустив її в 1991 році під назвою Python 0.9.0. [32] Python 2.0 був випущений в 2000 році і представив нові функції, такі як розуміння списків та система збору сміття з використанням підрахунку посилань, і був припинений з версією 2.7.18 у 2020 році [33]. Python 3.0 був випущений в 2008 році і був серйозним переглядом мови, яка не є повністю сумісною із зворотною суттю, і більша частина коду Python 2 не працює незміненою на Python 3.

Python стабільно входить до числа найпопулярніших мов програмування.

Python був задуманий наприкінці 1980-х Гвідо ван Россумом з Centrum Wiskunde & Informatica (CWI) в Нідерландах як спадкоємець мови програмування ABC, натхненної SETL, здатною обробляти винятки та взаємодіяти з Операційна система Amoeba. Його реалізація розпочалась у грудні 1989 р. Ван Россум взяв на себе виключну відповідальність за проект, як провідний розробник, до 12 липня 2018 року, коли він оголосив про свої "постійні канікули", виконуючи обов'язки як "Доброзичливий диктатор за життя" Пітона, титул, який йому надала спільнота Пітона, щоб відобразити його тривалий час. строкове зобов'язання як головний керівник проекту. Зараз він ділиться своїм керівництвом як член наглядової ради з п'яти осіб. У січні 2019 року активні розробники ядра Python обрали Бретта Кеннона, Ніка Коглана, Барі Варшаву, Керол Віллінг та Ван Россума членами керівної ради з п’яти членів. З тих пір Гвідо ван Россум відкликав свою кандидатуру на посаду Керівної ради 2020 року.

Python 2.0 був випущений 16 жовтня 2000 року з багатьма основними новими можливостями, включаючи збирач сміття, що виявляє цикл, та підтримку Unicode.

Python 3.0 був випущений 3 грудня 2008 року. Це був серйозний перегляд мови, який не є повністю сумісним із зворотною стороною. Багато його основних функцій були передані до версій Python 2.6.x та 2.7.x. Випуски Python 3 включають утиліту 2to3, яка автоматизує (принаймні частково) переклад коду Python 2 на Python 3.

Дата закінчення терміну служби Python 2.7 спочатку була встановлена на 2015 рік, а потім перенесена на 2020 рік через занепокоєння тим, що велика кількість існуючого коду не може бути легко перенесена на Python 3. Більше виправлень безпеки та інших удосконалень для нього випускати не буде. У кінці життя Python 2 підтримується лише Python 3.6.x і пізніші версії.

Python 3.9.2 та 3.8.8 були прискорені, оскільки всі версії Python мали проблеми із безпекою, що призвело до можливого віддаленого виконання коду та отруєння веб-кешу.

Python - мова програмування з багатьма парадигмами. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато його функцій підтримують функціональне програмування та аспектно-орієнтоване програмування (в тому числі метапрограмуванням та метаоб'єктами (магічні методи)). Багато інших парадигм підтримуються за допомогою розширень, включаючи дизайн за контрактом та логічне програмування.

Python використовує динамічне введення тексту та комбінацію підрахунку посилань та збирач сміття, що визначає цикл, для управління пам’яттю. Він також має динамічну роздільну здатність імен (пізнє прив'язування), яка зв'язує імена методів та змінних під час виконання програми.

Дизайн Python пропонує певну підтримку функціонального програмування за традицією Lisp. Він має функції фільтрування, картографування та зменшення; перелічити розуміння, словники, набори та вирази генераторів. Стандартна бібліотека має два модулі (itertools та functools), що реалізують функціональні інструменти, запозичені у Haskell та Standard ML.

Основна філософія мови узагальнена в документі Zen of Python (PEP 20), який включає такі афоризми, як:

- Красиве - краще, ніж потворне.
- Явне краще, ніж неявне.
- Просте - краще, ніж складне.
- Складний - це краще, ніж складний.
- Підрахунок читабельності.

Замість того, щоб усі його функціональні можливості були вбудовані в його ядро, Python був розроблений таким чином, щоб бути дуже розширюваним (з модулями). Ця компактна модульність зробила його особливо популярним як засіб додавання програмованих інтерфейсів до існуючих додатків. Бачення Ван Россума щодо малої основної мови з великою стандартною бібліотекою та легко розширюваним перекладачем випливало з його розчарувань у ABC, який підтримував протилежний підхід. Python прагне до більш простого, менш захараценого синтаксису та граматики, одночасно надаючи розробникам можливість вибору методології кодування. На відміну від девізу Perl "існує більше ніж один спосіб зробити це", Python охоплює "повинен бути один - і бажано лише один - очевидний спосіб зробити це" філософія дизайну. [69] Алекс Мартеллі, співробітник Фонду програмного забезпечення Python і автор книги про Python, пише, що "Описувати щось як" розумне "не вважається компліментом у культурі Python" .

Розробники Python прагнуть уникнути передчасної оптимізації та відкидають виправлення для некритичних частин еталонної реалізації CPython, які пропонують незначне збільшення швидкості ціною ясності. Коли швидкість важлива, програміст Python може переміщати критично важливі для часу функції до модулів розширення, написаних мовами, такими як C, або використовувати PyPy, своєчасний компілятор. Також доступний Cython, який перекладає скрипт Python на C і здійснює прямі виклики API рівня C в інтерпретатор Python.

Важливою метою розробників Python є підтримка його задоволення від використання. Це відображається в назві мови - данина поваги британській гумористичній групі Монті Пайтон - і в іноді грайливих підходах до навчальних посібників та довідкових матеріалів, таких як приклади, що стосуються спаму та яєць (із відомого етюду Монті Пайтона) стандартних foo and bar.

Поширеним неологізмом у спільноті Python є пітонічний, який може мати широкий діапазон значень, пов’язаних зі стилем програми. Сказати, що код пітонічний, означає сказати, що він добре використовує ідіоми Python, що він природний або демонструє вільну мову, що відповідає мінімалістичній філософії Python та наголосу на читабельності. На відміну від цього, код, який важко зрозуміти або читається як груба транскрипція з іншої мови програмування, називається непітонічним.

Користувачів та шанувальників Python, особливо тих, хто вважається обізнаним чи досвідченим, часто називають Pythonistas. Python має бути легкочитабельною мовою. Його форматування візуально не захаращене, і в ньому часто використовуються англійські ключові слова, де інші мови використовують розділові знаки. На відміну від багатьох інших мов, він не використовує фігурні дужки для розмежування блоків, а крапка з комою після операторів дозволена, але рідко, якщо взагалі використовується. Він має менше синтаксичних винятків та особливих випадків, ніж C або Pascal.

Python використовує пробіли, а не фігурні дужки або ключові слова, для відмежування блоків. Збільшення відступу відбувається після певних тверджень; зменшення відступу означає кінець поточного блоку. Таким чином, візуальна структура програми точно відображає семантичну структуру програми. Цю функцію іноді називають правилом "поза стороною", яке мають деякі інші мови, але в більшості мов відступ не має семантичного значення. Рекомендований розмір відступу - чотири пробіли.

Оператори Python включають (серед іншого):

- Оператор присвоєння, використовуючи один знак рівності =.

- Оператор if, який умовно виконує блок коду, разом з else та elif (скорочення else-if).
- Оператор for, який переглядає ітерабельний об’єкт, захоплюючи кожен елемент до локальної змінної для використання вкладеним блоком.
- Оператор while, який виконує блок коду, доки його умова відповідає істині.
- Оператор try, що дозволяє вилученням, що містяться у вкладеному блоці коду, ловити та обробляти за винятком речень; він також гарантує, що код очищення в блоці нарешті завжди буде виконуватися незалежно від того, як блок виходить.
- Оператор raise, що використовується для отримання вказаного винятку або повторного підняття спійманого винятку.
- Оператор class, який виконує блок коду та приєднує його локальний простір імен до класу для використання в об'єктно-орієнтованому програмуванні.
- Оператор def, що визначає функцію або метод.
- Оператор with з Python 2.5, випущений у вересні 2006 р. [82], який охоплює блок коду в контекстному менеджері (наприклад, отримання блокування перед запуском блоку коду та звільнення блокування після цього, або відкриття файлу, а потім закриваючи його), дозволяючи поведінку, схожу на отримання ресурсів - це ініціалізація (RAII), і замінює загальну ідіому try / нарешті.
- Оператор break, виходить із циклу.
- Оператор continue, пропускає цю ітерацію і продовжує наступний пункт.

- Оператор del видаляє змінну, що означає, що посилання з імені на значення видаляється, і спроба використання цієї змінної спричинить помилку. Видалену змінну можна перепризначити.
- Оператор pass, яка виконує функцію NOP. Це синтаксично потрібно для створення порожнього блоку коду.
- Оператор assert, який використовується під час налагодження для перевірки умов, які мають застосовуватися.
- Оператор yield, який повертає значення з функції генератора. З Python 2.5, yield також є оператором. Ця форма використовується для реалізації спільних програм.
- Оператор return, який використовується для повернення значення з функції.
- Оператор import, який використовується для імпорту модулів, функції чи змінні яких можна використовувати в поточній програмі. Є три способи використання імпорту: імпорт <ім'я модуля> [як <псевдонім>] або з <ім'я модуля> імпорт * або з <ім'я модуля> імпорт <визначення 1> [як <псевдонім 1>], <визначення 2> [як <псевдонім 2>], ...

Оператор присвоєння (=) діє шляхом прив'язки імені як посилання на окремий динамічно розподілений об'єкт. Потім змінні можуть бути відскочені в будь-який час до будь-якого об’єкта. У Python ім'я змінної є загальним власником посилання і не має фіксованого типу даних, пов’язаного з ним. Однак у даний момент часу змінна буде посилатися на якийсь об'єкт, який матиме тип. Це називається динамічним набором тексту і протиставляється статично набраним мовам програмування, де кожна змінна може містити лише значення певного типу.

Python не підтримує оптимізацію хвостових викликів або першокласні продовження, і, за словами Гвідо ван Россума, він ніколи не буде. Однак краща підтримка подібних до корутинів функціональних можливостей надається в 2.5, розширюючи генератори Python. До 2.5 генератори були ледачими ітераторами; інформація передавалась в односпрямованому напрямку з генератора. З Python 2.5 можна передавати інформацію назад у функцію генератора, а з Python 3.3 інформацію можна передавати через кілька рівнів стека.

Деякі вирази Python подібні до таких, що зустрічаються в таких мовах, як C та Java, а деякі - ні:

- Додавання, віднімання та множення однакові, але поведінка ділення відрізняється. У Python існує два типи поділів. Вони є розподілом підлоги (або цілочисельним поділом) // та плаваючою комою / поділом. Python також використовує оператор ** для піднесення до степені.
- З Python 3.5 був представлений новий оператор @ infix. Він призначений для використання бібліотеками, такими як NumPy, для множення матриць.
- З Python 3.8 був введений синтаксис: =, який називається «моржовим оператором». Він присвоює значення змінним як частину більшого виразу. [91]
- У Python == порівнює за значенням проти Java, яка порівнює числові значення за значенням, а об'єкти за посиланням. (Порівняння значень в Java на об'єктах можна виконувати методом equals ().) Оператор Python is is може використовуватися для порівняння ідентифікацій об'єктів (порівняння за посиланням). У Python порівняння можуть бути ланцюговими, наприклад a <= b <= c.
- Python використовує слова та, або, не для своїх булевих операторів, а не для символічного &&, ||,! використовується в Java та C.

- Python має тип виразу, що називається розумінням списку, а також більш загальний вираз, який називається генераторським виразом.
- Анонімні функції реалізовані за допомогою лямбда-виразів; однак вони обмежені тим, що тіло може бути лише одним виразом.
- Умовні вирази в Python пишуться як x, якщо c else y (відрізняється за порядком операндів від оператора c? X: y, загального для багатьох інших мов).
- Python розрізняє списки та кортежі. Списки записуються як [1, 2, 3], змінюються та не можуть використовуватися як ключі словників (ключі словника повинні бути незмінними в Python). Кортежі записуються як (1, 2, 3), є незмінними і, отже, можуть використовуватися як ключі словників, за умови, що всі елементи кортежу є незмінними. Оператор + може бути використаний для об'єднання двох кортежів, що безпосередньо не змінює їх вміст, а створює новий кортеж, що містить елементи обох передбачених кортежів. Таким чином, враховуючи змінну t, спочатку рівну (1, 2, 3), при виконанні t = t + (4, 5) спочатку обчислюється t + (4, 5), що дає (1, 2, 3, 4, 5) , який потім призначається назад до t, тим самим ефективно "модифікуючи вміст" t, одночасно відповідаючи незмінній природі об'єктів кортежу. Дужки не є обов’язковими для кортежів у однозначному контексті.
- Python має розпаковування послідовностей, при якому кілька виразів, кожен з яких обчислює будь-що, до чого можна присвоїти (змінну, властивість для запису тощо), пов’язані ідентично тому, що формує кортежні літерали, і, як ціле, розміщуються на ліва сторона знака рівності у твердженні про присвоєння. Оператор очікує ітерабельного об'єкта праворуч від знака рівності, який видає таку ж кількість значень, що і надані вираз для запису, коли його перебирають і буде перебирати його, присвоюючи кожне із створених значень відповідному виразу зліва .

- Python має оператор "формат рядка"%. Це функціонує аналогічно рядкам формату printf на C, наприклад "спам =% s яєць =% d"% ("бла", 2) оцінюється як "спам = бла-яйця = 2". У Python 3 та 2.6+ це було доповнено методом format () класу str, наприклад "спам = {0} яйця = {1}". формат ("бла", 2). Python 3.6 додав "f-рядки": blah = "blah"; яйця = 2; f'spam = {blah} яйця = {яйця} '.
- Рядки в Python можна об'єднати, "додавши" їх (той самий оператор, що і для додавання цілих чи значень з плаваючою точкою). Наприклад "спам" + "яйця" повертає "спамегги". Навіть якщо ваші рядки містять числа, вони все одно додаються як рядки, а не цілі числа. Наприклад "2" + "2" повертає "22".
- Python має різні типи рядкових літералів:
- Рядки, розділені одинарними або подвійними лапками. На відміну від оболонок Unix, мови Perl та Perl, що впливають на Perl, одинарні лапки та подвійні лапки функціонують однаково. Обидва типи рядків використовують зворотну скісну риску (\) як символ виходу. Інтерполяція рядків стала доступною в Python 3.6 як "відформатовані рядкові літерали".
  - Рядки з подвійними лапками, які починаються та закінчуються серією з трьох одинарних або подвійних лапок. Вони можуть охоплювати кілька рядків і функціонувати, як тут документи в оболонках, Perl і Ruby.
  - Сирі різновиди рядків, що позначаються префіксом рядкового літералу перед r. Послідовності втечі не інтерпретуються; отже, необроблені рядки корисні там, де буквенні зворотні слеші є загальними, такі як регулярні вирази та шляхи у стилі Windows. Порівняйте "@ -citation" у C #.
- Python має індекси масивів та вирази нарізування масивів у списках, що позначаються як [ключ], [старт: зупинка] або [старт: зупинка: крок]. Індекси базуються на нулі, а негативні індекси відносно кінця. Зрізи беруть елементи від початкового індексу до, але не включаючи, індексу зупинки. Третій параметр

зрізу, який називається кроком або кроком, дозволяє пропускати та обертати елементи. Індекси зрізів можуть бути опущені, наприклад, [:] повертає копію всього списку. Кожен елемент зрізу є неглибокою копією.

У Python чітко дотримується різниці між виразами та висловлюваннями, на відміну від таких мов, як Common Lisp, Scheme або Ruby. Це призводить до дублювання деяких функціональних можливостей.

Методи на об'єктах - це функції, приєднані до класу об'єкта; синтаксис instance.method (аргумент) - це для звичайних методів та функцій синтаксичний цукор для Class.method (екземпляр, аргумент). Методи Python мають явний параметр self для доступу до даних екземпляра, на відміну від неявного self (або цього) в деяких інших об’єктно-орієнтованих мовах програмування (наприклад, C ++, Java, Objective-C або Ruby).

Python використовує набір качок і має набрані об'єкти, але нетипізовані імена змінних. Обмеження типу не перевіряються під час компіляції; швидше, операції з об'єктом можуть зазнати невдачі, що означає, що даний об'єкт не є відповідним типом. Не дивлячись на те, що Python набирається динамічно, забороняє операції, які не є чітко визначеними (наприклад, додавання числа до рядка), а не намагається їх тихо зрозуміти.

Python дозволяє програмістам визначати власні типи за допомогою класів, які найчастіше використовуються для об'єктно-орієнтованого програмування. Нові екземпляри класів будуються за допомогою виклику класу (наприклад, SpamClass () або EggsClass ()), а класи є екземплярами типу метакласу (сам екземпляр самого себе), що дозволяє метапрограмувати та відображати.

До версії 3.0 Python мав два типи класів: старий і новий. Синтаксис обох стилів однаковий, різниця полягає в тому, чи об’єкт класу успадковується, прямо чи опосередковано (усі класи нового стилю успадковуються від об’єкта і є

екземплярами типу). У версіях Python 2, починаючи з Python 2.2, можна використовувати обидва типи класів. Класи старого стилю були ліквідовані в Python 3.0.

Довгостроковий план полягає в підтримці поступового набору тексту, а з Python 3.5 синтаксис мови дозволяє вказувати статичні типи, але вони не перевіряються в реалізації за замовчуванням, CPython. Експериментальна додаткова перевірка статичного типу з ім'ям mypy підтримує перевірку типу під час компіляції.

CPython - це посилальна реалізація Python. Він написаний на мові C, відповідаючи стандарту C89 з декількома вибраними функціями C99 (з випуском пізніших версій C він вважається застарілим; CPython включає власні розширення C, але розширення сторонніх розробників не обмежуються старими C версії, наприклад, можуть бути реалізовані з C11 або C ++). Він компілює програми Python у проміжний байт-код, який потім виконується його віртуальною машиною. CPython поширюється з великою стандартною бібліотекою, написаною на суміші C та рідного Python. Він доступний для багатьох платформ, включаючи Windows (починаючи з Python 3.9, інсталятор Python навмисно не вдається встановити в Windows 7 і 8; Windows XP підтримувався до Python 3.5) та більшості сучасних Unix-подібних систем, включаючи macOS (та Apple M1 Macs, починаючи з Python 3.9.1, з експериментальним інсталятором) та неофіційну підтримку, наприклад VMS. Переносимість платформи була одним із її перших пріоритетів, протягом періодів Python 1 та 2 підтримувались навіть OS / 2 та Solaris; підтримка відтоді відмовилася для багатьох платформ.

Розробка Python здійснюється в основному за допомогою процесу Програми вдосконалення Python (PEP), основного механізму пропонування

основних нових можливостей, збору вкладу спільноти з питань та документування рішень щодо проектування Python. Стиль кодування Python викладений у PEP 8. Видатні PEP переглядаються та коментуються спільнотою Python та керівною радою.

Покращення мови відповідає розробці посилальної реалізації CPython. Список розсилки python-dev є основним форумом для розвитку мови. Конкретні проблеми обговорюються у програмі відстеження помилок Roundup, розміщеній на bugs.python.org. Спочатку розробка здійснювалась у власному сховищі вихідного коду під управлінням Mercurial, поки Python не перейшов на GitHub у січні 2017 року.

Публічні випуски CPython бувають трьох типів, що відрізняються тим, яка частина номера версії збільшується:

Несумісні із зворотною мовою версії, де, як очікується, зламається код і його потрібно перенести вручну. Перша частина номера версії збільшується. Ці випуски трапляються рідко - версія 3.0 вийшла через 8 років після 2.0.

Основні або "функціональні" випуски відбувалися приблизно кожні 18 місяців, але з прийняттям щорічної каденції випуску, починаючи з Python 3.9, очікується, що це відбуватиметься раз на рік. Вони значною мірою сумісні, але вводять нові функції. Друга частина номера версії збільшується. Кожна основна версія підтримується виправленнями протягом декількох років після її випуску.

Випуски виправлень, які не вводять нових функцій, трапляються приблизно кожні 3 місяці і робляться, коли з останнього випуску виправлена достатня кількість помилок. Уразливості безпеки також виправлені у цих випусках. Третя і остання частина номера версії збільшується.

Багато альфа-, бета-версій та кандидатів на випуск також випускаються у вигляді попереднього перегляду та тестування перед остаточними випусками.

Хоча існує приблизний графік кожного випуску, вони часто затримуються, якщо код не готовий. Команда розробників Python контролює стан коду, запускаючи великий пакет модульних тестів під час розробки.

Основною академічною конференцією з Python є PyCon. Існують також спеціальні програми наставництва Python, такі як Pyladies.

Pythons 3.10 припиняє wstr (буде видалено в Python 3.12; це означає, що розширення Python потрібно буде змінити до того часу), а також планує додати відповідність шаблону до мови.

Використання

З 2003 р. Python стабільно входить до десятки найпопулярніших мов програмування в Індексі програмного забезпечення TIOBE, де, станом на лютий 2021 р., Це третя за популярністю мова (після Java та C). Він був обраний мовою програмування року (за "найвищий ріст рейтингу за рік") у 2007, 2010, 2018 та 2020 роках (єдина мова, яка зробила це чотири рази).

Емпіричне дослідження показало, що мови сценаріїв, такі як Python, є більш продуктивними, ніж звичайні мови, такі як C та Java, для проблем програмування, що включають маніпулювання рядками та пошук у словнику, і встановило, що споживання пам'яті часто "краще, ніж Java, а не набагато гірше, ніж C або C ++ ".

До великих організацій, які використовують Python, належать Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify та деякі менші організації, такі як ILM та ITA. Сайт соціальних новин Reddit був написаний переважно на Python.

Python може служити мовою сценаріїв для веб-додатків, наприклад, через mod_wsgi для веб-сервера Apache. Завдяки інтерфейсу шлюзу веб-сервера для полегшення роботи цих програм розроблено стандартний API. Веб-фреймворки,

такі як Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle і Zope, підтримують розробників у розробці та обслуговуванні складних додатків. Pyjs та IronPython можуть бути використані для розробки клієнтської частини програм на базі Ajax. SQLAlchemy може використовуватися як перетворювач даних у реляційну базу даних. Twisted - це фреймворк для програмування зв'язку між комп'ютерами, який використовується (наприклад) Dropbox.

Такі бібліотеки, як NumPy, SciPy та Matplotlib, дозволяють ефективно використовувати Python у наукових обчисленнях, а спеціалізовані бібліотеки, такі як Biopython та Astropy, забезпечують функціональність, що залежить від домену. SageMath - математичне програмне забезпечення з інтерфейсом ноутбука, програмоване на Python: його бібліотека охоплює багато аспектів математики, включаючи алгебру, комбінаторику, числову математику, теорію чисел і числення. OpenCV має палітурні прив'язки з багатим набором функцій для комп'ютерного зору та обробки зображень.

Python зазвичай використовується в проектах штучного інтелекту та машинному навчанні за допомогою таких бібліотек, як TensorFlow, Keras, Pytorch та Scikit-learn. Як мова сценаріїв з модульною архітектурою, простим синтаксисом та інструментами обробки багатого тексту, Python часто використовується для обробки природної мови. Python успішно вбудований у багато програмних продуктів як мову сценаріїв, включаючи програмне забезпечення методом скінченних елементів, таке як Abaqus, 3D-параметричний моделер, такий як FreeCAD, пакети 3D-анімації, такі як 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo , MotionBuilder, Softimage, композитор візуальних ефектів Nuke, програми для 2D-зображень, такі як GIMP, Inkscape, Scribus і Paint Shop Pro, та музичні нотатні програми, такі як автори та капели. Налагоджувач GNU використовує Python як гарний принтер для показу складних

структур, таких як контейнери C ++. Esri пропагує Python як найкращий вибір для написання сценаріїв в ArcGIS. Він також використовувався в декількох відеоіграх і був прийнятий як перша з трьох доступних мов програмування в Google App Engine, інші дві - Java і Go.

Багато операційних систем включають Python як стандартний компонент. Він постачається з більшістю дистрибутивів Linux, AmigaOS 4 (з використанням Python 2.7), FreeBSD (як пакет), NetBSD, OpenBSD (як пакет) та macOS і може використовуватися з командного рядка (терміналу). Багато дистрибутивів Linux використовують інсталятори, написані на Python: Ubuntu використовує інсталятор Ubiquity, тоді як Red Hat Linux і Fedora використовують інсталятор Anaconda. Gentoo Linux використовує Python у своїй системі управління пакунками Portage.

Python широко використовується в галузі інформаційної безпеки, в тому числі в розробці експлойтів.

Більшість програм Sugar для одного ноутбука на дитину XO, які зараз розробляються в Sugar Labs, написані на Python. Проект одноплатної комп'ютерної програми Raspberry Pi прийняв Python як основну мову програмування користувачів.

LibreOffice включає Python і має намір замінити Java на Python. Його постачальник сценаріїв Python є основною функцією з версії 4.0 від 7 лютого 2013 року.

Виходячи зі сфери використання і величезного функціоналу мови Python, який був описаний вище, можна зробити висновок про його мультифункціональність. Саме через це дана мова програмування була обрана для виконання даної роботи.

3.1.2. Фреймворк Django

Django - це безкоштовний веб-фреймворк на основі Python, що відповідає архітектурному зразку model-template-views (MTV). Він підтримується Django Software Foundation (DSF), американською незалежною організацією, створеною як некомерційна організація 501 (c) .

Основна мета Django - полегшити створення складних веб-сайтів, керованих базами даних. Структура підкреслює багаторазовість та "підключеність" компонентів, менше коду, низьку зв'язок, швидкий розвиток та принцип "не повторюватися". Python використовується всюди, навіть для налаштувань, файлів та моделей даних. Django також надає додатковий адміністративний інтерфейс створення, читання, оновлення та видалення, який генерується динамічно за допомогою самоаналізу та налаштовується за допомогою моделей адміністратора.

Деякі добре відомі сайти, які використовують Django, включають PBS, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, і Nextdoor.

3.2. Перевірка розподіленої комп’ютерної мережі підприємства легкої промисловості

Використання розробленої розподіленої комп’ютерної мережі підприємства легкої промисловості є доцільним, тому що:

- був розроблений та створений сайт для роботи з мережею
- до сайту підключена база даних, в якій зберігаються всі дані
- база даних зберігається в файлах проекту на локальному сервері, що дозволить легко перенести її при побудові на хості.

При початку роботи з сайтом відкривається сторінка авторизації (рис. 3.1), яка дозволяє перейти на головну (рис. 3.2) або до сторінки реєстрації (рис. 3.3).

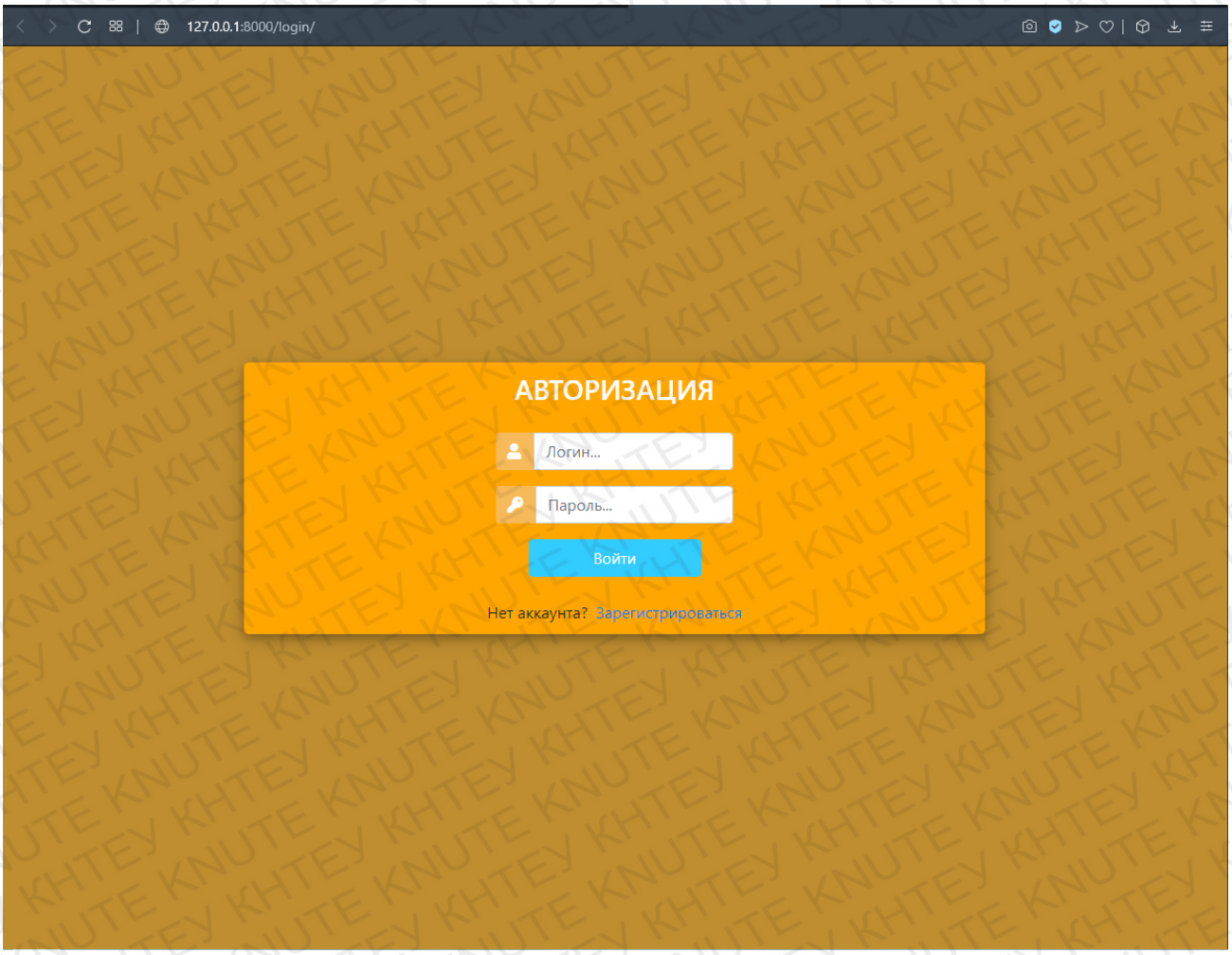

Рис. 3.1 — Форма авторизації

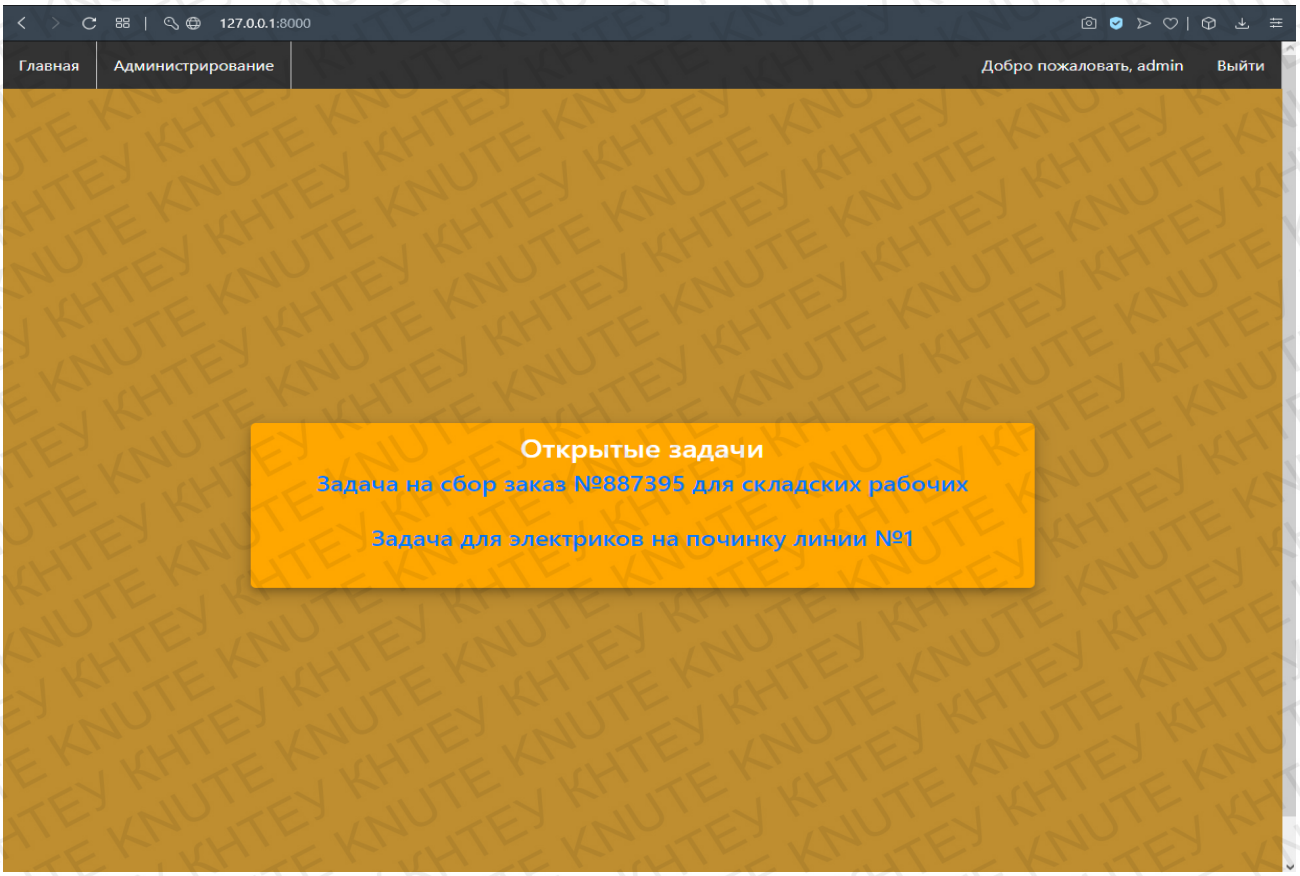

Рис. 3.2 — Головна сторінка

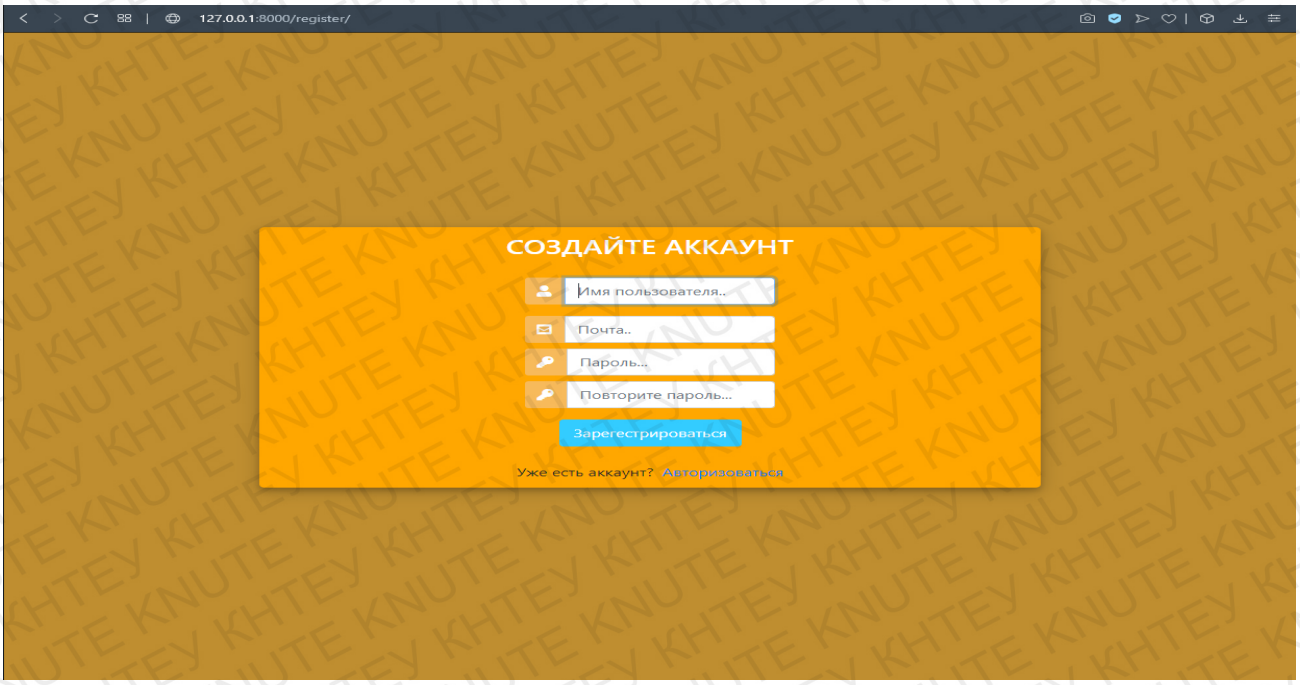

Рис. 3.3 — Форма реєстрації

З головної сторінки користувач має змогу перейти до подання звіту по конкретній задачі (рис. 3.4) або (за наявності прав адміністратора) перейти в меню адміністрування (рис. 3.5).

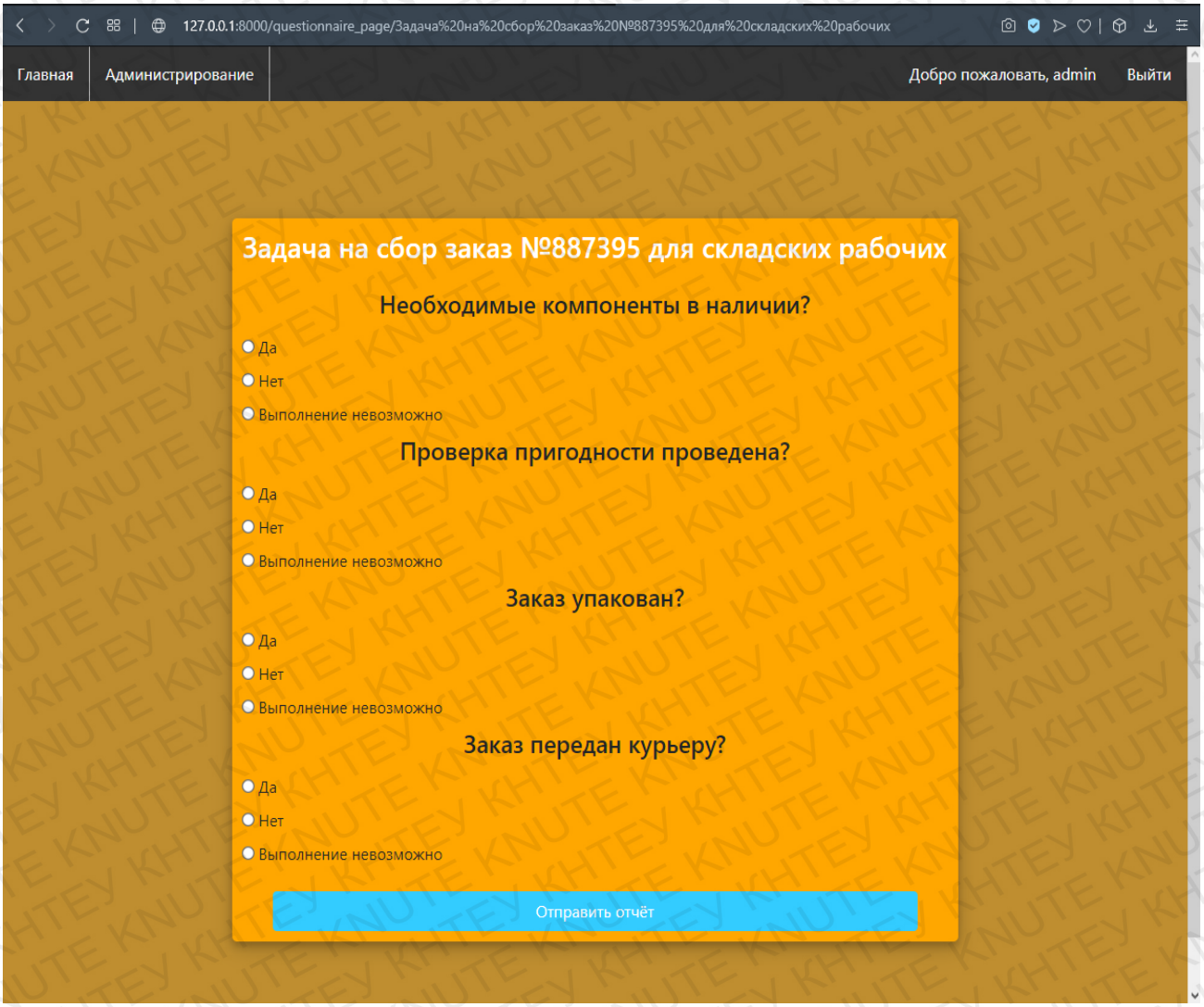

Рис. 3.4 — Форма звіту по задачі

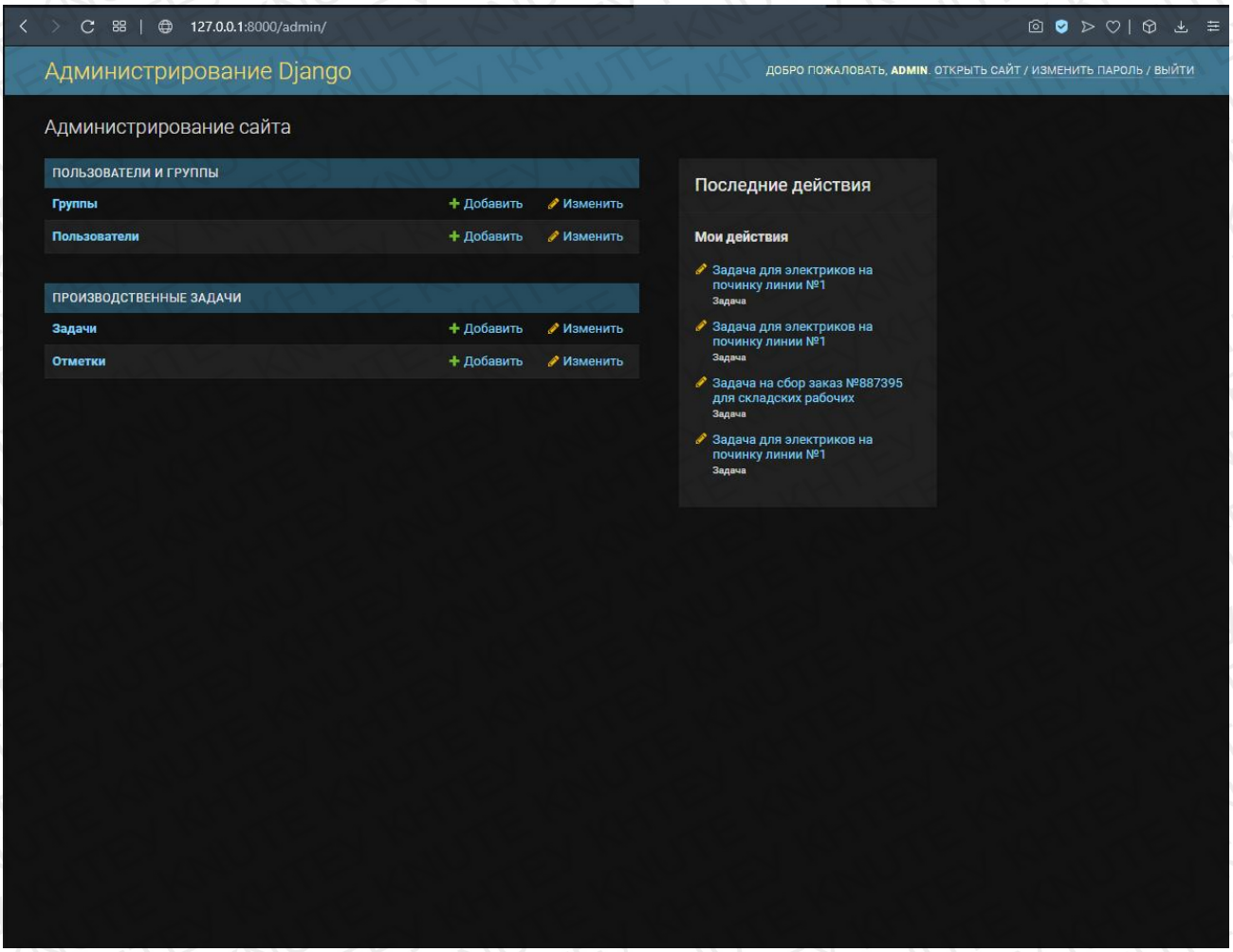

Рис. 3.5 — Форма адміністрування

ВИСНОВКИ

Узагальнення результатів теоретичної та дослідно-експериментальної роботи дає підстави для таких висновків:

1. Питання розробки розподіленої комп'ютерної мережі підприємств легкої промисловості, представлені в наукових дослідженнях, неоднозначні за рахунок великого різноманіття технологій розробки та можливого функціоналу. Розвиток технологій створення розподілених мереж підприємств стимулює щоденне зростання обсягів інформації, необхідних для повноцінного функціонування того чи іншого підприємства;
2. Розроблено модель розподіленої комп’ютерної мережі підприємства легкої промисловості, яка складається з модулів користувача і адміністратора, кожен з яких має свій функціонал, який в сукупності дає повноцінну систему обробки внутрішніх задач підприємства;
3. З’ясовано, що використання програмно-технічних засобів розподілених комп'ютерних мереж підприємства легкої промисловості має свої особливості, наприклад необхідність оновлення даних «в рантаймі», необхідність створення ролей користувачів для достатнього рівню контролю, необхідність побудови власної серверної частини для забезпечення конфіденційності і безпеки інформації;
4. Розробка розподіленої комп'ютерної мережі підприємств легкої промисловості буде ефективною при застосуванні технологій, що забезпечують збирання, передачу, обробку, відображення і зберігання інформації розподіленої комп'ютерної мережі підприємств легкої промисловості, дають змогу користувачам різних ролей мати доступ лише

до тих даних, які націлені на дану групу користувачів та забезпечать достатній рівень взаємодії між користувачами і системою;

Проведене дослідження не вичерпує всіх питань розробки розподіленої комп'ютерної мережі підприємств легкої промисловості. Перспективним є подальше дослідження з таких напрямів, як:

1. Додавання нових типів взаємодії в систему;
2. Додавання нових ролей в систему;
3. Додавання можливості зберігання внутрішніх документів та документообігу в цілому;
4. Створення механізму контролю якості виробництва продукції;
5. Розробка рекомендацій щодо застосування розподіленої комп'ютерної мережі підприємств легкої у сфері легкої промисловості України де подано технологію обробки внутрішніх задач підприємства його робітниками;

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tanenbaum, Andrew S.; Steen, Maarten van (2002). Distributed systems: principles and paradigms. Upper Saddle River, NJ: Pearson Prentice Hall. ISBN 0-13-088893-1.

2. Andrews (2000). Dolev (2000). Ghosh (2007), p. 10.

3. Magnoni, L. (2015). "Modern Messaging for Distributed Sytems (sic)". Journal of Physics: Conference Series. 608 (1): 012038. doi:10.1088/1742-6596/608/1/012038. ISSN 1742-6596.

4. Godfrey (2002).

5. Andrews (2000), p. 291–292. Dolev (2000), p. 5.

6. Lynch (1996), p. 1.

7. Ghosh (2007), p. 10.

8. Andrews (2000), pp. 8–9, 291. Dolev (2000), p. 5. Ghosh (2007), p. 3. Lynch (1996), p. xix, 1. Peleg (2000), p. xv.

9. Andrews (2000), p. 291. Ghosh (2007), p. 3. Peleg (2000), p. 4.

10. Ghosh (2007), p. 3–4. Peleg (2000), p. 1.

11. Ghosh (2007), p. 4. Peleg (2000), p. 2.

12. Ghosh (2007), p. 4, 8. Lynch (1996), p. 2–3. Peleg (2000), p. 4.

13. Lynch (1996), p. 2. Peleg (2000), p. 1.

14. Ghosh (2007), p. 7. Lynch (1996), p. xix, 2. Peleg (2000), p. 4.

15. Ghosh (2007), p. 10. Keidar (2008).

16. Lynch (1996), p. xix, 1–2. Peleg (2000), p. 1.

17. Peleg (2000), p. 1.

18. Papadimitriou (1994), Chapter 15. Keidar (2008).

19. See references in Introduction.

20. Bentaleb, A.; Yifan, L.; Xin, J.; et al. (2016). "Parallel and Distributed Algorithms" (PDF). National University of Singapore. Retrieved 20 July 2018.

21. Andrews (2000), p. 348.

22. Andrews (2000), p. 32.

23. Peter (2004), The history of email.

24. Banks, M. (2012). On the Way to the Web: The Secret History of the Internet and its Founders. Apress. pp. 44–5. ISBN 9781430250746.

25. Tel, G. (2000). Introduction to Distributed Algorithms. Cambridge University Press. pp. 35–36. ISBN 9780521794831.

26. Ohlídal, M.; Jaroš, J.; Schwarz, J.; et al. (2006). "Evolutionary Design of OAB and AAB Communication Schedules for Interconnection Networks". In Rothlauf, F.; Branke, J.; Cagnoni, S. (eds.). Applications of Evolutionary Computing. Springer Science & Business Media. pp. 267–78. ISBN 9783540332374.

27. "Real Time And Distributed Computing Systems" (PDF). ISSN 2278-0661. Retrieved 2017-01-09.

28. Vigna P, Casey MJ. The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order St. Martin's Press January 27, 2015 ISBN 9781250065636

29. Hieu., Vu, Quang (2010). Peer-to-peer computing : principles and applications. Lupu, Mihai., Ooi, Beng Chin, 1961-. Heidelberg: Springer. p. 16. ISBN 9783642035135. OCLC 663093862.

30. Lind P, Alm M (2006), "A database-centric virtual chemistry system", J Chem Inf Model, 46 (3): 1034–9, doi:10.1021/ci050360b, PMID 16711722.

31. Chiu, G (1990). "A model for optimal database allocation in distributed computing systems". Proceedings. IEEE INFOCOM'90: Ninth Annual Joint Conference of the IEEE Computer and Communications Societies.

Elmasri & Navathe (2000), Section 24.1.2.