Назва дипломної: **Розробка системи міграції даних**

**Анотація.** Коротенький зміст основних результатів роботи (2-3 речення) українською і англійською мовами.

**Вступ**

В якому наводиться ґрунтовний опис:

*Проблематики та актуальності* теми роботи, *аналіз існуючих підходів* і методів розв’язання та практичного застосування результатів їх розв’язання ( при цьому **обов’язкові посилання на використані інформаційні джерела**(монографії, статті, Інтернет інформація)), мета роботи, предмет дослідження. Об’єкт дослідження, аргументація новизни і коротка ідея запропонованого Вами підходу.

**Розділ 1. Постановка задачі**

* 1. Основні означення , теореми,леми, твердження, гіпотези.

**Advanced Message Queuing Protocol**

Відкритий стандарт протоколу прикладного рівня для проміжного програмного забезпечення, орієнтованого на обробку повідомлень. Семантика обміну повідомленнями налаштовується під потреби конкретного проекту. «Брокер повідомлень», який здійснює маршрутизацію, зазвичай гарантує доставку, розподіл потоків даних, підписку на потрібні типи повідомлень.  
Ідея AMQP була розроблена в 2003 році Джон О'Хара з JPMorgan Chase в Лондоні. JPMorgan Chase уклав контракт з iMatix Corporation, з середини 2004 до середини 2006 року, на розробку C-брокера і його протоколу. У 2005 році JPMorgan Chase звернувся до інших фірм, щоб сформувати робочу групу, яка включала Cisco Systems, Айона Technologies, iMatix, Red Hat і Transaction Workflow Innovation Standards Team (TWIST). У тому ж році JPMorgan Chase укладає партнерство з Red Hat, для створення Apache Qpid, спочатку на Java, а потім і на C++.

**Принцип роботи AMQP**

«Повідомлення» (message) відправляються в «точку обміну». Точка обміну розподіляє повідомлення в одну або кілька «черг». При цьому в точці обміну повідомлення не зберігаються. Точки обміну бувають трьох типів:

* розгалуження ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *fanout*) - повідомлення передається в усі причеплені до неї черги ;
* прямі (direct) - повідомлення передається в чергу з ім'ям, що збігається з ключем маршрутизації (routing key), який вказується при відправці повідомлення ;
* теми (topic ) - щось середнє між розгалуженням і прямим типом, повідомлення передається в черзі, для якої збігається [маска](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D1%80%D0%B0%D0%B7) на ключ маршрутизації.

Повідомлення зберігаються в чергах доти, поки не буде забрано клієнтом. Клієнт завжди забирає повідомлення з однієї або декількох черг.

**RabbitMQ**

Однією з найвідоміших систем, що реалізують AMQP є така системи, як RabbitMQ.

RabbitMQ - платформа, яка реалізує систему обміну повідомленнями між компонентами програмної системи (Message Oriented Middleware) на основі стандарту AMQP (Advanced Message Queuing Protocol) [3]. RabbitMQ випускається під Mozilla Public License. RabbitMQ створений на основі випробуваної Open Telecom Platform, що забезпечує високу надійність і продуктивність промислового рівня і написана на мові Erlang.

RabbitMQ складається з:

* RabbitMQ Сервера
* Підтримка протоколи HTTP, XMPP і STOMP
* Клієнтських бібліотек AMQP для Java і .NET Framework
* Різних плагінів (плагін моніторингу та управління через HTTP або веб-інтерфейс, або плагін «Shovel» для передачі повідомлень між брокерами)

Підтримується горизонтальне масштабування для побудови кластерної архітектури.

Як движок бази даних, для зберігання повідомлень, використовується Mnesia.

**Основні поняття RabbitMQ**

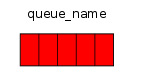
Producing – процес надсилання повідомлень.

Producer – програма, що надсилає повідомлення.



Producer

Queue– черга, яка живе в середині RabbitMQ. Черзі обов'язково повинно бути просвоєнно ім'я (queue\_name).



Queue

Consumer – програма, яка очікує та приймає повідомлення.



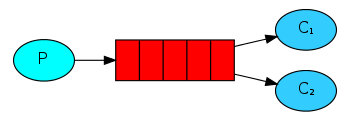
Consumer

Стандартна архітектура системи на базі RabbitMQ, виглядає наступним чином



Producer надсилає повідомлення в чергу. Consumer слухає чергу і отримує повідомлення з неї.

Зазвичай в реальних проектах використовується наступна архітектура



Producer надсилає повідомлення в чергу. Consumer1, Consumer2 і ConsumerN слухають чергу і один з них, або всі отримають повідомлення. Який з Consumer отримає повідомлення залежить від налаштування RabbitMQ, варіцій таких налаштувань надзвичайно багато.

**Протокол транспортного рівня TCP**

Протокол TCP реалізує потокову модель передачі інформації, хоча в його основі, як і в основі протоколу UDP, лежить обмін інформацією через пакети даних. Він являє собою орієнтований на встановлення логічного зв'язку (орієнтований на з'єднання), надійний (забезпечує перевірка контрольних сум, передача підтвердження в разі правильного прийому повідомлення, повторна передачі пакета даних в разі неотримання підтвердження протягом певного проміжку часу, правильна послідовність отримання інформації, повний контроль швидкості передачі даних) двобічний спосіб зв'язку між процесами в мережі.

**TCP-сегмент**

Блок даних TCP називається сегментом, хоча часто також використовують слово пакет, але таке вживання може вносити плутанину з IP-пакетом.

TCP-сегмент складається із TCP-заголовка і поля Дані (Data), яке називають сегментом даних або пейлодом або SDU.

Стандартний розмір TCP-заголовка — 20 байт, але з використанням опцій розмір може зростати до 60 байт. Як правило, опціями хости обмінюються на етапі встановлення з'єднання.

Розмір сегменту даних (поля даних) визначається опцією MSS (Максимальний розмір сегменту, Maximum segment size) на етапі встановлення з'єднання. Якщо обміну опціями не відбулося, то розмір сегменту даних встановлюється за замовчуванням 536 байт.[9] Розмір сегменту даних тісно пов'язаний з MTU (Максимальний блок передачі). Фактично MSS дорівнює MTU з відніманням розміру IP- і TCP-заголовків. Наприклад у сучасній мережі Ethernet MTU дорівнює 1500 байт; тоді оптимальний розмір MSS буде 1460 байтів (1500 мінус 20 байт заголовка IP і 20 байт заголовка TCP).



**WebSocket**

WebSocket — це протокол, що призначений для обміну інформацією між браузером та веб-сервером в режимі реального часу. Він забезпечує двонаправлений повнодуплексний канал зв'язку через один TCP-сокет. WebSocket спроектовано для втілення у веб-браузерах та веб-серверах, але може також використовуватись будь-якою клієнт-серверною системою. Прикладний програмний інтерфейс WebSocket був стандартизований W3C, крім того протокол WebSocket стандартизований IETF як RFC 6455

Перед встановленням з’єднання поверх TCP, відбувається «рукопотискання» (handshake), поверх HTTP:

GET /ws HTTP/1.1

Host: pmx

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Version: 6

Sec-WebSocket-Origin: [http://pmx](http://pmx/)

Sec-WebSocket-Extensions: deflate-stream

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Відповідь сервера

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=

Клієнт надсилає ключ Sec-WebSocket-Key, що є випадковим значенням, закодованим з використанням base64. Щоб сформувати відповідь, до закодованого ключа додають рядок [GUID](https://uk.wikipedia.org/wiki/GUID) 258EAFA5-E914-47DA-95CA-C5AB0DC85B11, а потім (не розкодовуючи ключ) хешують через [SHA1](https://uk.wikipedia.org/wiki/SHA1) та base64. І на завершення, результуючу відповідь вкладають в заголовок Sec-WebSocket-Accept.

WebSocket підтримують такі браузери:

* Google Chrome (починаючи з версії 4.0.249.0);
* Apple Safari (починаючи з версії 5.0.7533.16);
* Mozilla Firefox (починаючи з версії 4);
* Opera (починаючи з версії 10.70 9067);

**Електронна пошта**

Електро́нна по́шта або е-пошта — спосіб обміну цифровими повідомленнями між людьми використовуючи цифрові пристрої, такі як комп'ютери та мобільні телефони, що робить можливим пересилання даних будь-якого змісту (текстові документи, аудіо-, відеофайли, архіви, програми).

Електронна пошта схожа на звичайну пошту. Звичайний лист складається із конверта, на якому зазначена адреса отримувача і стоять штампи поштових відділень шляху слідування, та вмісту — власне листа. Електронний лист складається із заголовків, які містять службову інформацію (про автора листа, отримувача, шлях проходження листа), які служать, умовно кажучи, конвертом, та власне вміст самого листа. За аналогією зі звичайним листом, відповідним методом можна внести в електронний лист інформацію якого-небудь іншого роду, наприклад, фотографію тощо. Як і у звичайному листі можна поставити свій підпис. Звичайний лист може не дійти до адресата або дійти з запізненням, — аналогічно і електронний лист. Звичайний лист доволі дешевий, а електронна пошта — найдешевший вид зв'язку.

Адреса містить ім’я комп’ютера та ім’я конкретного користувача цього комп’ютера, якому адресоване повідомлення.

Електронні скриньки розміщуються на поштових серверах. Для кожної скриньки на поштовому сервері відводиться спеціальне місце. На одному поштовому сервері не може бути двох скриньок з однаковими назвами.

Електронна адреса складається з двох частин, розділених знаком @ Наприклад:

**alexander.mykulych@gmail.com**

alexander.mykulych – ім’я скриньки, gmail.com – доменне ім’я поштового сервера, на якому знаходиться ця скринька. Повідомлення, що надіслані за цією адресою, будуть передані на комп’ютер з іменем gmail.com користувачу на ім’я alexander.mykulych.

**Протоколи передачі електронної пошти**

* IMAP
* POP3
* SMTP
* UUCP

**Веб-сервіс (Web Service)**

Веб-сервіс – програмна система, яка ідентифікується веб-адресою, з стандартизованим інтерфесом.

Веб-сервіси можуть взаємодіяти один з одним і зі сторонніми додатками за допомогою повідомлень, заснованих на певних протоколах (SOAP, XML-RPC, REST і т.д.). Веб-служба є одиницею модульності при використанні сервіс-орієнтованої архітектури.

**Telegram**

Telegram — програмне забезпечення для смартфонів на базі Android та iOS, яке дозволяє обмінюватися текстовими повідомленнями та різноманітними файлами, зокрема графічними файлами та відеофайлами, а також безкоштовно телефонувати іншим користувачам програми.

Підтримує німецьку, англійську, іспанську, італійську, нідерландську, португальську, арабську та китайську. Українська мова не доступна за замовчанням, але її можна встановити власноруч. Додаток має версії для ПК з операційними системами Linux, Windows та OS X.

Обліковий запис користувача прив'язуються до номера мобільного телефона: щоб авторизуватися він повинен ввести код авторизації з отриманного СМС-повідомлення. Надісланий код має обмежений термін придатності, після спливання якого код стає неактуальним. Таким чином, користувач позбавляється необхідності запам'ятовувати чи зберігати десь свій пароль.

Станом на вересень 2016 року месенджером «Telegram» користуються 100 мільйонів осіб.

За допомогою спеціального API сторонні розробники можуть створювати «ботів», спеціальні акаунти, керовані програмами. Типові боти відповідають на спеціальні команди в персональних і групових чатах, також вони можуть здійснювати пошук в інтернеті або виконувати інші завдання, застосовуються задля розваг або в бізнесі.

**Elasticsearch**

Elasticsearch — вільне програмне забезпечення, пошуковий сервер, розроблений на базі Lucene. Надає розподілений, мультиарендний повнотекстовий пошуковий рушій з HTTP веб-інтерфейсом і підтримкою безсхемних JSON документів.

Є найбільш популярним пошуковим рушієм, випереджаючи Apache Solr. Розробляється на Java і випускається під ліцензією Apache License.

Elasticsearch може використовуватись для індексування та пошуку будь-яких типів документів. Він надає масштабовний пошук, має пошук близький до реального часу і підтримку мультиарендності.

Elasticsearch має можливість розподілення, індекси можуть бути розділені по шардах, при чому кожен шард може мати нуль чи більше реплік. Кожен вузол містить один чи більше шардів і діє як координатор делегування операцій на потрібний шард. Балансування та маршрутизація виконується автоматично.

Приклад використання

* Реалізація пошуку по веб-сайту, наприклад пошук товарів в інтернет-магазині. В цьому випадку Elasticsearch індексує каталоги товарів, та надає можливості пошуку та припущення щодо автозаповнення.
* Зберігання журналів подій чи транзакцій, аналізування і добування даних для отримання тенденцій, статистик, висновків, аномалій. В даному випадку можна використовувати Logstash для збору, об'єднання, аналізу даних, і потім перенаправляти ці дані в Elasticsearch для подальшого опрацювання.
* Для розробки, наприклад, платформи по ціновому оповіщенню, що дозволяє досвіченим в цій сфері користувачам визначати правила типу «Я зацікавлений в придбанні електронного ґаджету XXX і я хочу бути оповіщеним якщо ціна у будь-якого постачальника впаде нижче XXX протягом наступного місяця». В даному випадку можна збирати ціни, індексувати їх в Elasticsearch і використовувати функцію зворотнього пошуку: зіставляти коливання цін з запитом користувача і при відповідності до запиту надсилати сповіщення.
* Для впровадження аналітики/Бізнес-аналітики в проект, коли треба швидко досліджувати, аналізувати, візуалізувати надзвичайно великі об'єми даних (мільйони чи мільярди записів). В даному випадку доцільно використання Elasticsearch для збереження даних і Kibana для побудови користувацьких панелей відображення і візуалізації необхідних аспектів. До того ж, можна використовувати агрегаційні функції Elasticsearch для здійснення комплексної бізнес-аналітики даних.

**Kibana**

Kibana - це система з відкритим кодом, для візуалізації даних, плагін для Elasticsearch. Він забезпечує можливість візуалізації даних, які індексуються на кластері Elasticsearch.

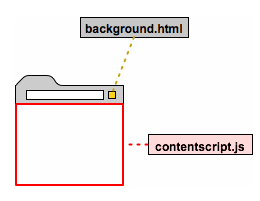
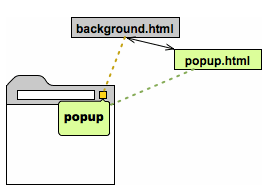
**Розширення Google Chrome**

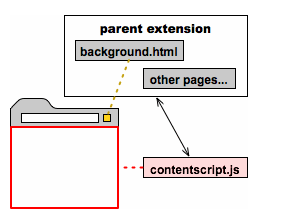
Розширення Google Chrome - це розширення браузера, що змінюють та модифікують його роботу. Розширення написані з використанням веб-технологій, таких як HTML, JavaScript і CSS.

В фізичному представленні розширення – це архів файлів: HTML, CSS, JavaScript, зображень та конфігураційних файлів. Розширення можуть використовувати всі інтерфейси API, що браузер надає для веб-сторінок, від XMLHttpRequest та JSON до HTML5.

В браузері Chrome розширення мають наступну архітектуру:

1. background page
2. UI pages (popup.html)
3. Content scripts
4. Other pages





Background Page

Html сторінка, яка містить JavaScript код, для контролю логіки розширення. Зазвичай вона має назву background.html і ділиться на два типи: «повністю фонова сторінка» (persistent background page) та «сторінка подій» (event page). Різниця між типами полягає в тому, що «persistent background page» завжди знаходиться у відкритому режимі, а «event page» переходить у потрібні режими тільки при певних подіях, таких як: натискання іконки розширення, чи завершення роботи браузера.

UI pages (popup.html)

Сюди входять звичайні html-сторінки, які містить розширення. Прикладом є сторінка, яка появляється при активації розширення (popup.html). Таких сторінок може бути необмежена кількість. Їх завдання надавати користувачу інтерфейс для роботи з функціями розширення або надавати певну інформацію, кожна сторінка має можливість спілкуватись з фоновою сторінкою (background page), передаючи їй потрібну інформацію, та отримуючи відповідь.

Content scripts

До даного типу відноситься js-файл, який забезпечує інтерактивність з веб-сторінкою, яка відкрита в браузері. Браузер забезпечує завантаження даного файла в контексті основної веб-сторінки після чого файл має можливість довантажувати інші файли, модифікувати сторінку та файли, які забезпечують роботу сторінки.

Other pages

До цього типу відносяться усі html-файли, які використовуються фоновою сторінкою чи контент-скріптом.

**Chrome DevTools Protocol**

Chrome DevTools Protocol дозволяє додатку оперувати, досліджувати, дебажити і профілювати Chrome, Chromium і інші Blink-based браузери.

Інструментарій Chrome DevTools Protocol включає в себе певну кількість доменів (DOM, Debugger, Network etc.). Домен – це інструментальна одиниця, яка дозволяє отримувати інформацію про певний аспект роботи браузера в реальному часі. Кожен домен визначає методи і події, які генерують ці методи. Методи і події подаються в серіалізованому представлені JSON-об’єктів, з фіксованою структурою.

Для даної роботи детальніше розглянемо домен – Network, який дає дозволяє відслідковувати мережеву активність веб-сторінки, та надає інформацію про http, файли, дані і інші запити, та відповіді, їхні заголовки і тіла, час і т.п.

Основні методи, які надає домен:

|  |  |
| --- | --- |
| Network.enable | Дозволяє відстежувати мережу, мережеві події будуть відправлені клієнту |
| Network.disable | Вимикає відслідковування активності в мережі, запобігає відправці мережевих подій клієнту. |

Події, які відповідають методу Network.enable:

|  |  |
| --- | --- |
| Network. resourceChangedPriority | Активується, якщо при завантаженні ресурсів пріоритет змінюється |
| Network. requestWillBeSent | Активуэться, коли сторінка збирається послати запит HTTP |
| Network. requestServedFromCache | Активується, якщо запит закінчив завантаження з кеша |
| Network. responseReceived | Активується, коли HTTP відповідь доступна |
| Network. loadingFinished | Виникає, коли запит HTTP завершить завантаження |
| Network. loadingFailed | Виникає, коли запит HTTP не вдалося завантажити |
| Network. webSocketWillSendHandshakeRequest | Виникає, коли WebSocket збирається ініціювати рукопотискання |
| Network. webSocketHandshakeResponseReceived | Активується, коли відповідь рукопотискання закінчено успішно |
| Network. webSocketClosed | Активується, після створення WebSocket |