

Збір вимог та оцінка навантаження

Конструювання програмного забезпечення

Ігор Мірошниченко
КНУ імені Тараса Шевченка, ФІТ

Конструювання програмного забезпечення

ЗМІСТ

- Про мене
- DataCamp Group
- Збір вимог до системи
- Функціональні вимоги
- Нефункціональні вимоги
- Приклади
- Розрахунок навантаження на систему
- Приклади розрахунків

ПРО МЕНЕ

- Мірошниченко Ігор Вікторович
- кандидат економічних наук, доцент
- доцент кафедри технологій управління, ФІТ, КНУ імені Тараса Шевченка
- викладач Міжнародного інституту бізнесу (МВА)

✉ ihor.miroshnychenko@kneu.ua

📠 Data Mirosh

linkedin [@ihormiroshnychenko](https://www.linkedin.com/in/ihormiroshnychenko)

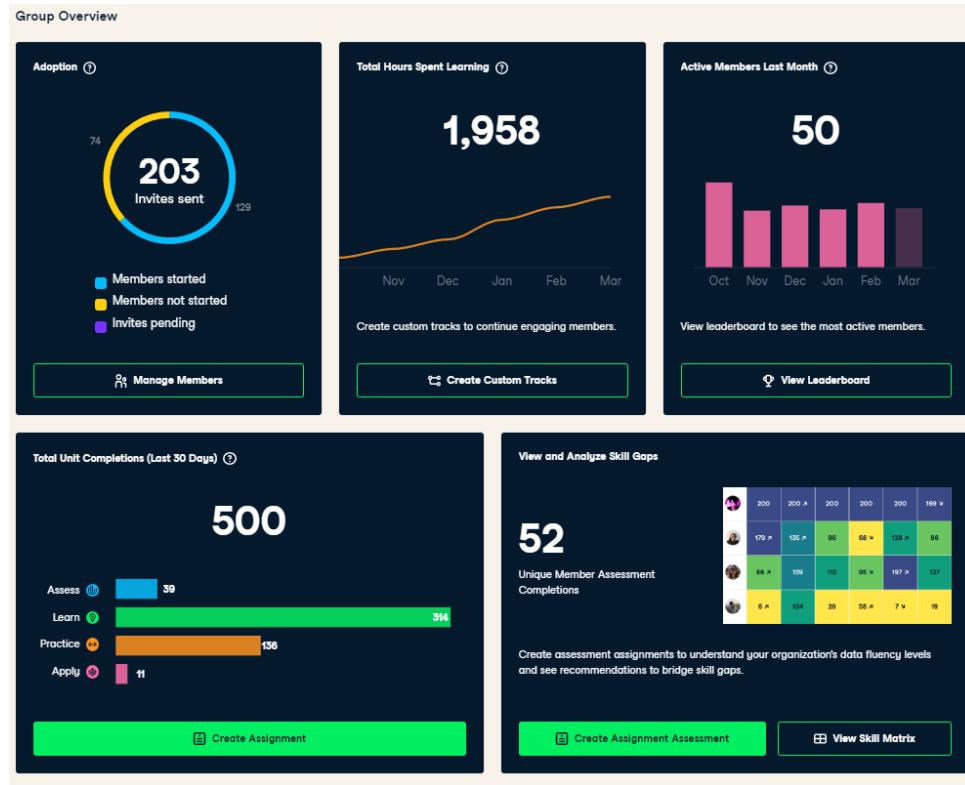
github [@aranaur](https://github.com/aranaur)

🏡 aranaur.rbind.io

DATA CAMP GROUP



DATA CAMP GROUP



Career Tracks

Our career tracks are hand-picked by industry experts. You will learn all you need to start a new career in the data science field.

All Power BI Python R SQL Tableau

② Which technology should I choose?

R Programmer	Data Scientist with R
④ 48 hours ⑧ 12 courses	④ 88 hours ⑧ 22 courses
Data Analyst with R	Python Programmer
④ 36 hours ⑧ 9 courses	④ 67 hours ⑧ 17 courses
Data Scientist with Python	Data Analyst with Python
✓ Track Completed	④ 36 hours ⑧ 9 courses

PREPARE FOR CERTIFICATION

DATA CAMP GROUP



Як долучитися?

1. Приєднатися до телеграм-каналу Data Mirosh
2. Зареєструйтесь на DataCamp
3. Приєднайтесь до класу за посиланням

Примітка

Клас буде активний **з 11 жовтня 2023 року до 11 квітня 2023 року**, після чого буде відкрито наступний потік. Слідкуйте за оновленнями.

ЗБІР ВИМОГ ДО СИСТЕМИ

Конструювання програмного забезпечення

ЗБІР ВИМОГ ДО СИСТЕМИ

Оскільки задача проєктування систем майже завжди формується у відкритій формі, нам потрібно окреслити функціонал та визначити вимоги до системи.

Слід почати з пошуку відповідей на наступні питання:

- **Функціональні вимоги:** які основні можливості дає система користувачам?
- **Нефункціональні вимоги:** наскільки надійною та швидкою має бути система?
- **Обмеження функціоналу:** які обмеження на функціонал системи накладаються зовнішніми факторами?

ФУНКЦІОНАЛЬНІ ВИМОГИ

ФУНКЦІОНАЛЬНІ ВИМОГИ

Функціональні вимоги - це можливості, які система надає користувачеві.

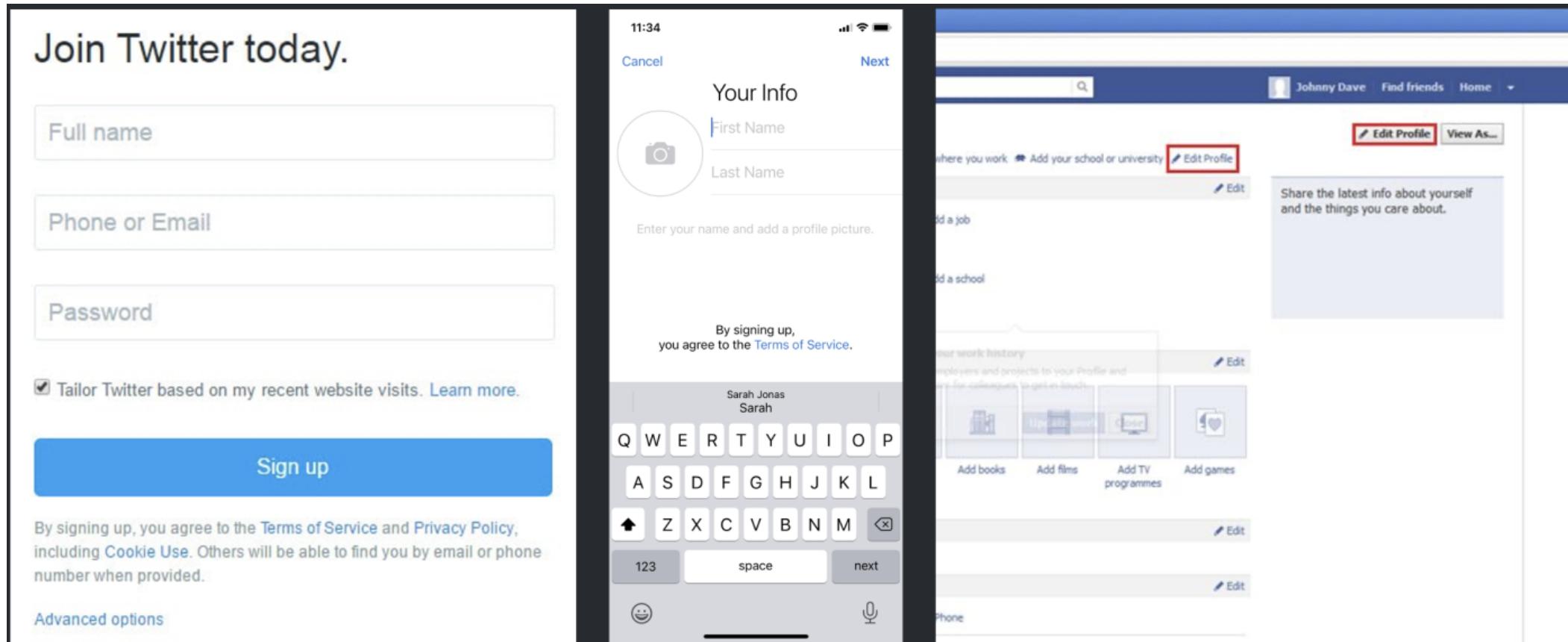
Наприклад, для соціальних мереж це:

- додавання в друзі;
- спілкування між користувачами;
- створення пабліків.

Для цього варто виписати найвідоміші властивості системи в порядку важливості та вказати, над якими ми плануємо сконцентруватися надалі.

ПРИКЛАД СОЦІАЛЬНИХ МЕРЕЖ

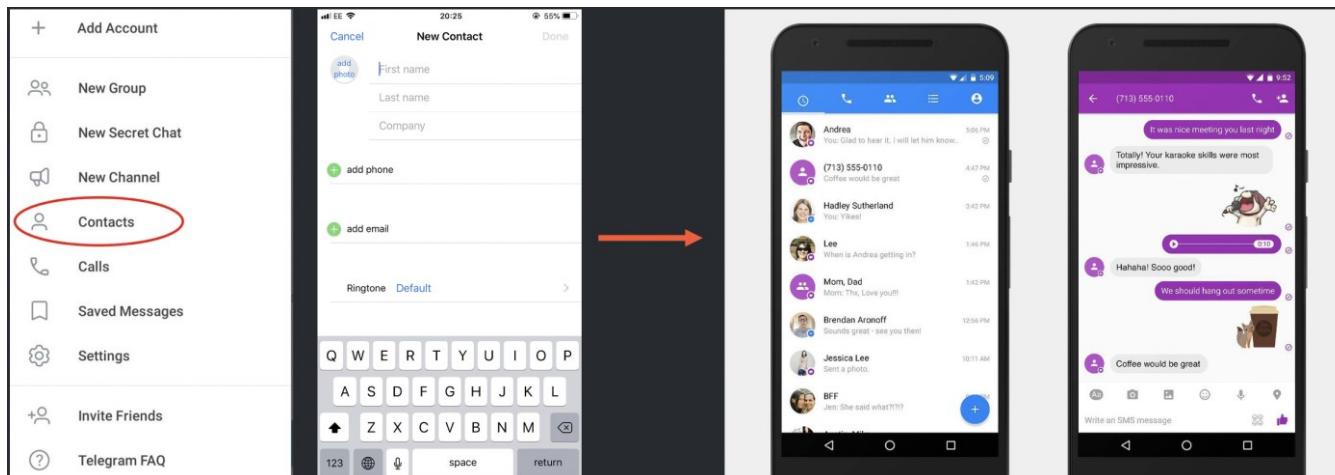
- Створення профілю (додавання опису і фото)



ПРИКЛАД СОЦІАЛЬНИХ МЕРЕЖ

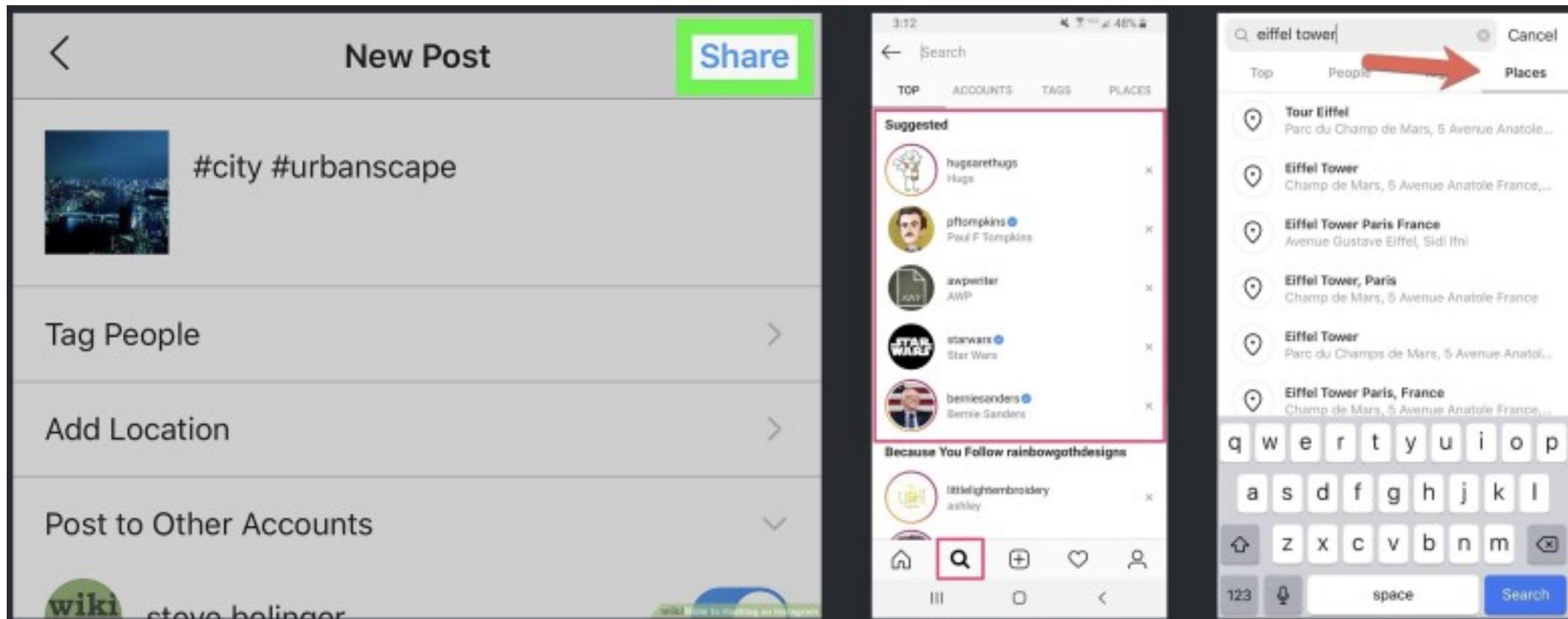
Потім, залежно від спрямованості соціальної мережі, можемо представити такі функції, як:

- створення контенту (публікації, фото, відео, історії)
- відстеження контенту (по користувачам чи тегам?)
- додавання контактів з мережі (номер телефону?)
- спілкування зі своїми контактами (текстові повідомлення, відеодзвінки, голосові повідомлення?)



ПРИКЛАД СОЦІАЛЬНИХ МЕРЕЖ

Якісь соцмережі можуть підтримувати всі напрямки і додаткові можливості, на кшталт додавання геотегів, пошуку контенту, рекомендацій тощо.



ФУНКЦІОНАЛЬНІ ВИМОГИ

Основні думки для створення соціальної мережі

Має сенс вибрати той функціонал, на якому зупинимося:

- створення профілю з юзернеймом;
- створення тільки текстових постів;
- підписка на пости інших користувачів.

А інші можливості згадати, але винести за рамки проектованої версії:

- додавання контактів за номером телефону;
- спілкування зі своїми контактами з відеодзвінками;
- додавання тегів і геоміток до постів;
- повнотекстовий пошук контенту;
- відображення постів від користувачів поблизу та ін.



НЕФУНКЦІОНАЛЬНІ ВИМОГИ

Конструювання програмного забезпечення

НЕФУНКЦІОНАЛЬНІ ВИМОГИ

Нефункціональні вимоги - це те, як має поводитися система в роботі.

Типові запитання, на які хочемо відповісти:

- Чи має система бути абсолютно надійною або якісь дані можуть загубитися?
- Що нам важливіше - швидка відповідь системи чи гарантія врахування останніх змін?
- З якою затримкою можуть застосовуватися внесені нами зміни?
- З якою затримкою відображаються запитувані дані?

Changes in the Admin console or to Google services

Changing settings or performing tasks, such as managing mobile devices and users, will take effect in a few minutes for most customers. Some changes can take up to 24 hours to take effect. Sometimes, you'll see a warning message if there's an expected delay.

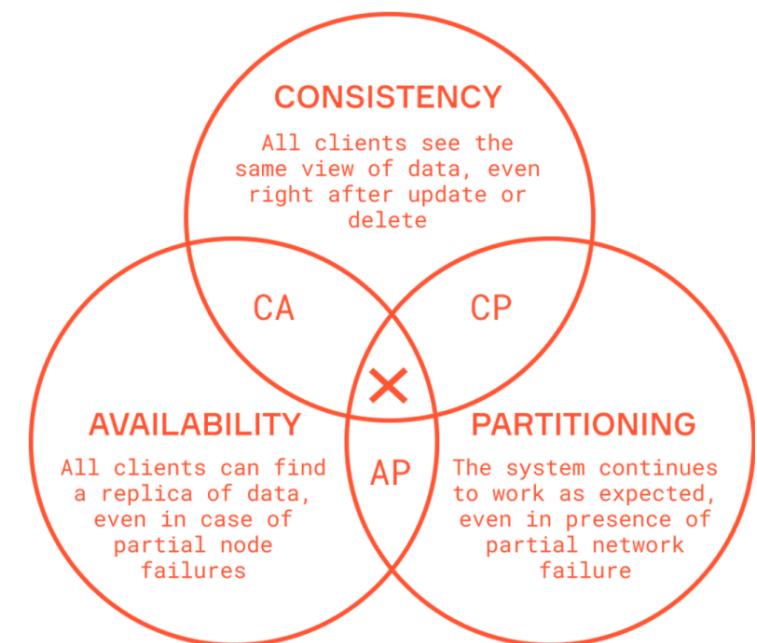
НЕФУНКЦІОНАЛЬНІ ВИМОГИ

На перший погляд всі ці питання можуть виглядати дивними — чому б не зробити систему якомога швидшою та надійнішою? Але відповіді на них допоможуть нам зрозуміти, які компроміси ми готові робити.

Наприклад, **CAP-теорема** говорить, що система не може одночасно бути:

- Consistent (консистентною),
- Available (доступною)
- Partition-tolerant (стійкою до розбиття на частини).

Тому в залежності від вимог до системи ми можемо вибрати, які з цих властивостей нам важливіші.



НЕФУНКЦІОНАЛЬНІ ВИМОГИ

Для **соціальної мережі** важливо, щоб вона завжди була **доступною**.

Соціальна мережа також розділена, маючи користувачів по всьому світу (і сервери, відповідно).

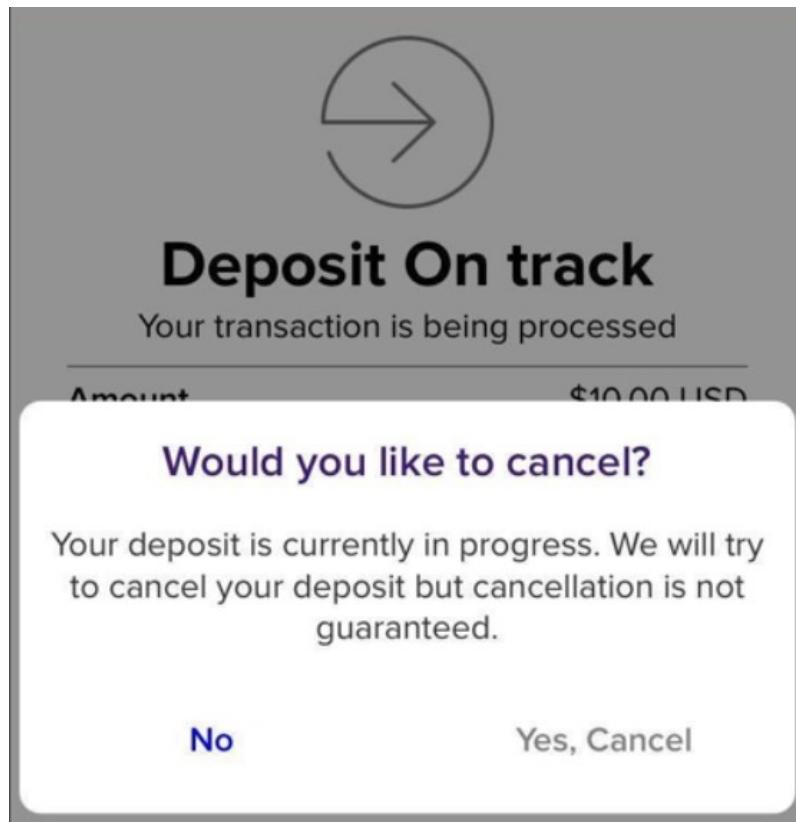
Однак, якщо **користувач публікує пост**, можливо, ми **не одразу** його побачимо.

Нічого страшного, якщо станеться затримка, рано чи пізно він все одно стане доступним для всіх користувачів. Тобто, **консистентність не така критична**.

НЕФУНКЦІОНАЛЬНІ ВИМОГИ

У випадку з **банківським додатком** успішно проведена грошова транзакція повинна означати надходження і видимість коштів на рахунку адресата, ніяк інакше.

Ціною цьому може бути **недоступність застосунку** якийсь час.



ПРИКЛАДИ

Конструювання програмного забезпечення



СЕРВІС КОРОТКИХ ПОСИЛАНЬ

Навіщо потрібен такий сервіс?

Щоб перетворювати довгі посилання на короткі, зручні для обміну.

MTeams.link
Here is the list of links you have created on MTeams.link

Recently created(s)

 <https://mteams.link/24L4>
Link : https://teams.microsoft.com/l/meetup-join/19%3ameeting_MjhhZWNjYTktZjMwYS00MWZjLWFlymMtZjBkj15ODA3NWE2%40thread.v2/0?context=%7b%22Tid%22%3a%22726204ed-52fe-41de-b46b-6041c7cb52ef%22%2c%22Oid%22%3a%2233180b7f-ebe0-4136-aabd-6a9afe96c5be%22%7d
Expiration : 03/10/2024
Click : 3

 <https://mteams.link/4RB2>
Link : https://teams.microsoft.com/l/meetup-join/19%3ameeting_Y2IzMWE2ZjctYjY0My00YjilWEyNGQtNjc2M2E0NDcxMmFm%40thread.v2/0?context=%7b%22Tid%22%3a%22726204ed-52fe-41de-b46b-6041c7cb52ef%22%2c%22Oid%22%3a%2233180b7f-ebe0-4136-aabd-6a9afe96c5be%22%7d
Expiration : 03/10/2024
Click : 1

 <https://mteams.link/DGNL>
Link : https://teams.microsoft.com/l/meetup-join/19%3ameeting_NGE2YWFiOTctYjFkMi00NWY5LWE5NjEtMGQ3NzdjOWZjMDRj%40thread.v2/0?context=%7b%22Tid%22%3a%22726204ed-52fe-41de-b46b-6041c7cb52ef%22%2c%22Oid%22%3a%2233180b7f-ebe0-4136-aabd-6a9afe96c5be%22%7d
Expiration : 03/10/2024
Click : 2

СЕРВІС КОРОТКИХ ПОСИЛАНЬ

Функціональні вимоги:

- користувач може ввести довге посилання та отримати коротке;
- коли користувач переходить по короткому посиланню, він перенаправляється на вихідне посилання.
- користувач може вибрати коротке посилання самостійно.
- посилання “протухає” через певний час.

Нефункціональні вимоги:

- система повинна бути доступною.
- перенаправлення з мінімальною затримкою.
- коротке посилання має бути унікальним.
- коротке посилання має бути випадковим, щоб його не було легко підібрати.

Бонус: збір аналітики, API для розробників.

ХОСТИНГ ТЕКСТІВ

Навіщо потрібен такий сервіс?

Щоб ділитися кодом і шукати в ньому помилки разом.

PasteBin API TOOLS FAQ + paste Search... Raw

Darts Tournament
SPOCOMAN OCT 6TH, 2023 174 NEVER ADD COMMENT

SHARE TWEET

Not a member of Pastebin yet? [Sign Up](#), it unlocks many cool features!

Python 0.58 KB | None | 0 raw download clone embed print report

```

1. startPoints = int(input())
2. moves = 0
3.
4. while startPoints > 0:
5.     command = input()
6.     moves += 1
7.
8.     if command == "bullseye":
9.         break
10.
11.    points = int(input())
12.
13.    if command == "double ring":
14.        points *= 2
15.    elif command == "triple ring":
16.        points *= 3
17.
18.    startPoints -= points
19.
20. if startPoints == 0:
21.     print(f"Congratulations! You won the game in {moves} moves!")
22. elif startPoints < 0:
23.     print(f"Sorry, you lost. Score difference: {abs(startPoints)}")
24. else:
25.     print(f"Congratulations! You won the game with a bullseye in {moves} moves!")
26.
```

europe.r

```

1. # data from http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/population-distribution-demography/geostat
2. # Originally seen at http://spatial.ly/2014/08/population-lines/
3.
4. # So, this blew up on both Reddit and Twitter. Two bugs fixed (southern Spain was a mess,
5. # and some countries where missing -- measure twice, submit once, dammit), and two silly superfluous lines removed after
6. # @hadleywickham pointed that out. Also, switched from geom_segment to geom_line.
7.
8. # The result of the code below can be seen at http://imgur.com/ob8c8ph
9.
10 library(tidyverse)
11
12 read_csv('../data/geostat-2011/GEOSTAT_grid_POP_1K_2011_V2_0_1.csv') %>%
13   rbind(read_csv('../data/geostat-2011/JRC-GHSI_AIT-grid-POP_1K_2011.csv')) %>%
14   mutate(TOT_P_CON_DF='')
15   mutate(lat = as.numeric(gsub('.*N([0-9]+)[EW].*', '\\\\1', GRD_ID))/100,
16         lng = as.numeric(gsub('.*[EW]([0-9]+)', '\\\\1', GRD_ID)) * ifelse(gsub('.*([EW].*', '\\\\1', GRD_ID) == 'W', -1, 1) / 100))
17   filter(lng > 25, lng < 60) %>%
18   group_by(lat=round(lat, 1), lng=round(lng, 1)) %>%
19   summarize(value = sum(TOT_P, na.rm=TRUE)) %>%
20   ungroup() %>%
21   complete(lat, lng) %>%
22   ggplot(aes(lng, lat + 5*(value/max(value, na.rm=TRUE)))) +
23     geom_line(size=0.4, alpha=0.8, color="#5A3E37", aes(group=lat), na.rm=TRUE) +
24     gthemes::theme_map() +
25     coord_equal(0.9)
26
27 ggsave('/tmp/europe.png', width=10, height=10)
```

 sj-porter-dev commented on Apr 23, 2017

Love the simplicity and readability of this sample. Thank you for sharing.

Pastebin

ХОСТИНГ ТЕКСТІВ

Функціональні вимоги:

- Користувач може ввести текст і отримати посилання на нього.
- Підсвічування тексту.
- Посилання “протухають” через певний час.
- Можливість редагування тексту.

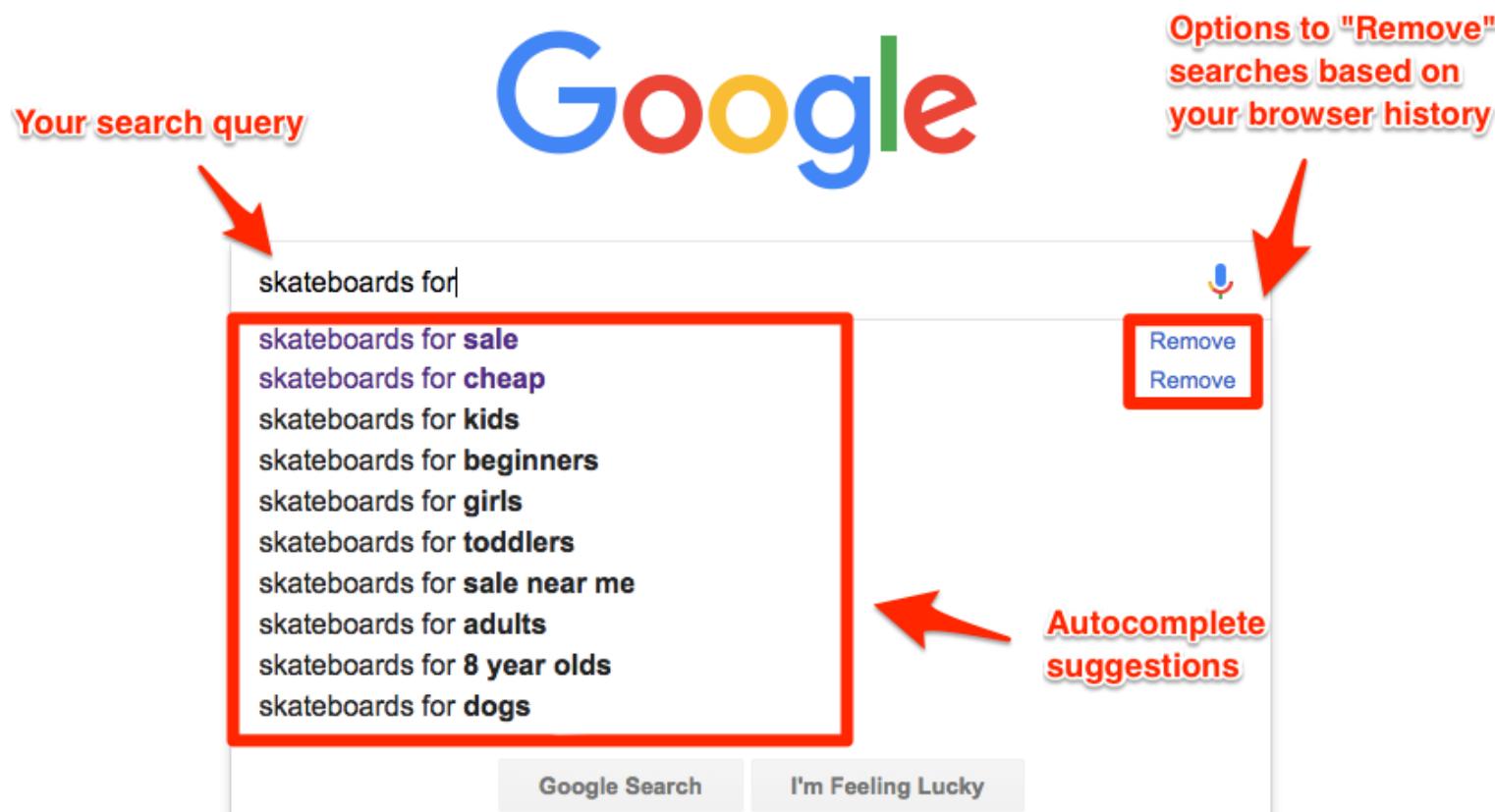
Нефункціональні вимоги:

- Висока надійність сервісу - завантажені дані не повинні зникати.
- Висока доступність сервісу - інакше доступ до потрібних даних пропадає.
- Збережені тексти повинні відображатися з мінімальною затримкою.
- Короткі посилання мають бути випадкові настільки, щоб їх не можна було підібрати.

АВТОДОПОВНЕННЯ

Навіщо потрібен такий сервіс?

Щоб при введенні перших букв запиту пропонувалися варіанти доповнення.



АВТОДОПОВНЕННЯ

Функціональні вимоги:

- Користувач може ввести початок запиту і отримати список варіантів доповнення.
- Варіанти оновлюються по мірі введення користувачем тексту.

Нефункціональні вимоги:

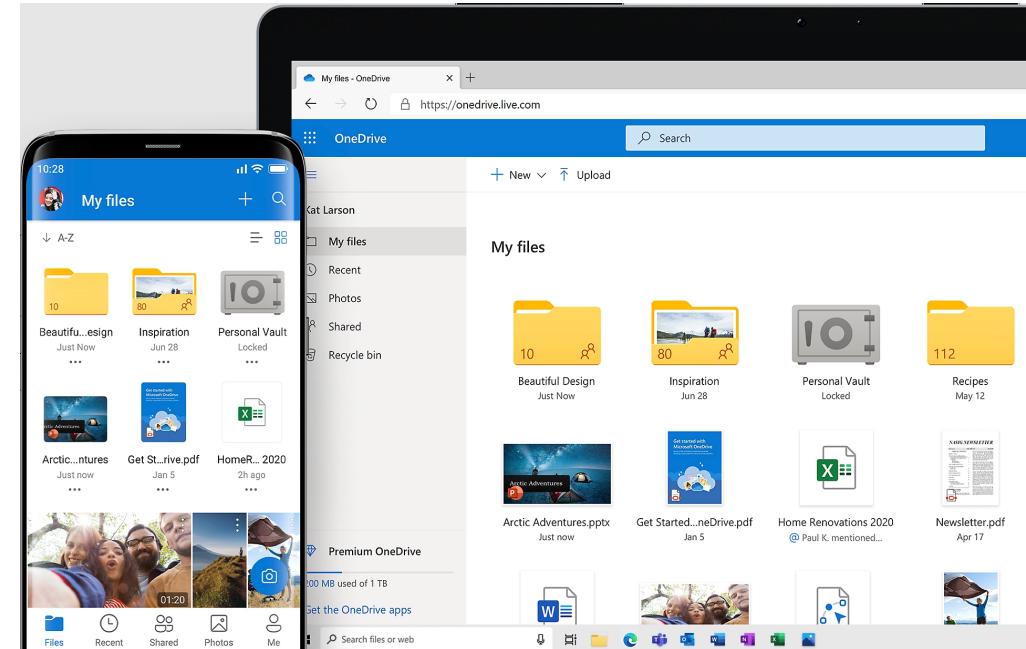
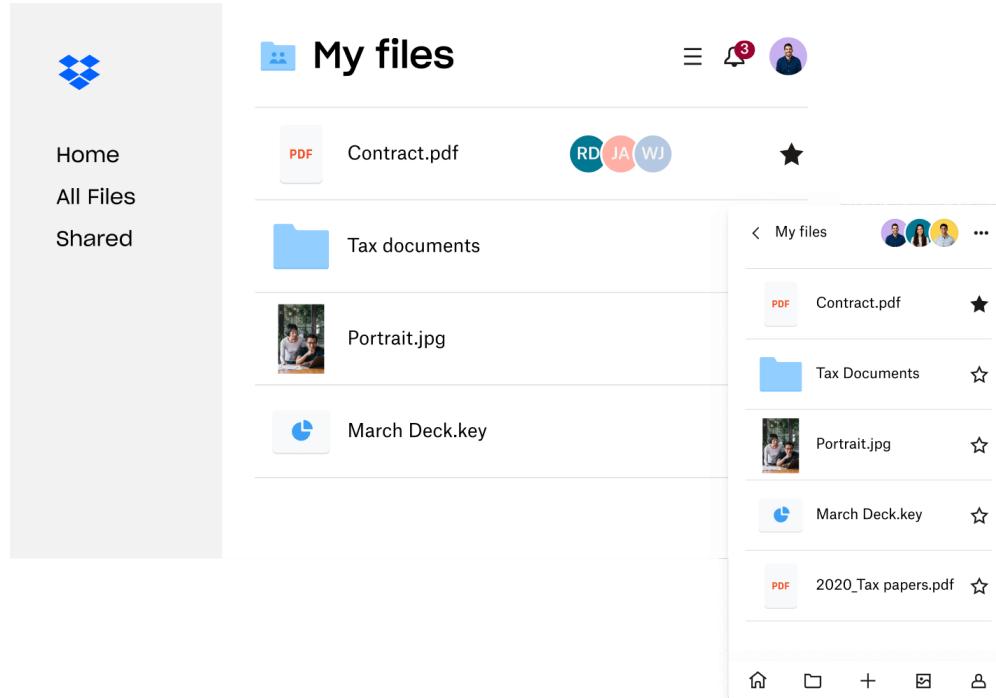
- Висока швидкість відображення варіантів доповнення.

Бонус: враховувати минулу історію запитів, профіль користувача, контекст тощо.

ХМАРНИЙ ДИСК

Навіщо потрібен такий сервіс?

Щоб зберігати файли в хмарі та мати до них доступ з різних пристроїв.



ХМАРНИЙ ДИСК

Функціональні вимоги:

- Користувач може завантажити та вивантажити файли з гаджетів.
- Користувач може ділитися файлами з іншими користувачами.
- Автоматична синхронізація між пристроями.
- Можна редагувати файл офлайн, синхронізація відбувається після підключення до мережі.

Нефункціональні вимоги:

INSTAGRAM

Навіщо потрібен такий сервіс?

Щоб ділитися фотографіями їжі та себе у дзеркалі.



INSTAGRAM

Функціональні вимоги:

- Користувачі можуть завантажувати та переглядати фотографії.
- Користувачі можуть шукати інші фотографії за описом.
- Користувачі можуть підписуватися на фотографії один одного.
- Є стрічка з популярних фото всіх відстежуваних користувачів

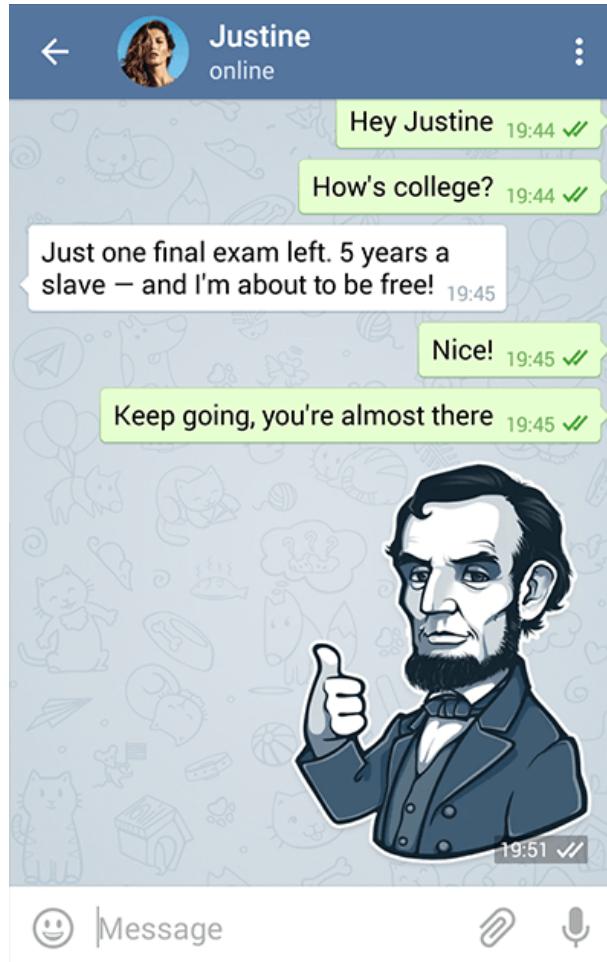
Нефункціональні вимоги:

- Висока доступність сервісу - інакше користувачі перейдуть у Snapchat або TikTok.
- Стрічка з фотографіями повинна відображатися за сотні мілісекунд (інакше див. п. 1).
- Консистентність не така критична - підписники скількись потерплять без ваших селфі.
- Висока надійність сервісу - жодне селфі або фото їжі не повинно загубитися.

TELEGRAM

Навіщо потрібен такий сервіс?

Щоб від спілкування Вас відволікали роботою.



TELEGRAM

Функціональні вимоги:

- Користувачі можуть відправляти один одному повідомлення.
- Користувачі можуть створювати канали для розсилки повідомлень.
- Користувачі можуть відправляти один одному файли.

Нефункціональні вимоги:

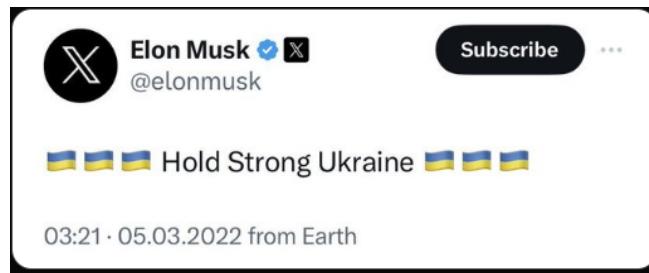
- Спілкування відбувається в реальному часі з мінімальними затримками.
- Консистентність — однакові повідомлення відображаються у всіх чатах.
- Висока надійність — повідомлення не повинні загубитися.
- Висока доступність — інакше користувачі перейдуть у Viber або WhatsApp.

Бонус: стікери, групові чати, сторіз 🎨, тощо

TWITTER

Навіщо потрібен такий сервіс?

Щоб почитати чергову маячню Маска і що з цього приводу думає цивілізований світ.



Elon Musk verified X
@elonmusk · Начать читать

X

Ukraine-Russia Peace:

- Redo elections of annexed regions under UN supervision. Russia leaves if that is will of the people.
- Crimea formally part of Russia, as it has been since 1783 (until Khrushchev's mistake).
- Water supply to Crimea assured.
- Ukraine remains neutral.

Yes	40.9 %
No	59.1 %

2 748 378 голосов · Конечные результаты

7:15 PM · 3 окт. 2022 г. ⓘ

95,2 тыс. Ответить ⬆ Поделиться Читать 111,9 тыс. ответов

Volodymyr Zelenskyy / Володимир Зеленський verified X
@ZelenskyUa

Which @elonmusk do you like more?

One who supports Ukraine	78.8%
One who supports Russia	21.2%

2,431,641 votes · Final results

9:45 PM · Oct 3, 2022

TWITTER

Функціональні вимоги:

- Користувачі можуть додавати короткі пости.
- Пости можуть містити фото або відео.
- Користувачі можуть відстежувати пости інших.
- Є стрічка з твітів усіх відстежуваних користувачів.

Нефункціональні вимоги:

- Висока доступність сервісу - інакше користувачі повернуться в живий журнал.
- Стрічка з твітами має відображатися за сотні мілісекунд (інакше див. п. 1).
- Консистентність не така критична - підписники трошки потерплять без ваших думок.

Бонус: пошук, реплаї, гарячі теми, повідомлення, рекомендації, аватарки з NFT мавпами.

NETFLIX

Навіщо потрібен такий сервіс?

Щоб подивитися Гаррі Поттера перед Різдвом.



Конструювання програмного забезпечення

NETFLIX

Функціональні вимоги:

- Користувачі можуть переглядати фільми та серіали.
- Користувачі можуть шукати контент за назвою.
- Сервіс відстежує оцінки та статистику переглядів.
- Користувачам рекомендуються нові фільми та серіали

Нефункціональні вимоги:

- Висока доступність сервісу - інакше користувачі будуть дивитися котиків у конкурентів.
- Висока чуйність сервісу - відео повинні програватися без підвисань.
- Консистентність не така критична - після тижневого очікування одна хвилина непомітна.

Бонус: жанри, популярне, вибране, списки на подивитися потім.

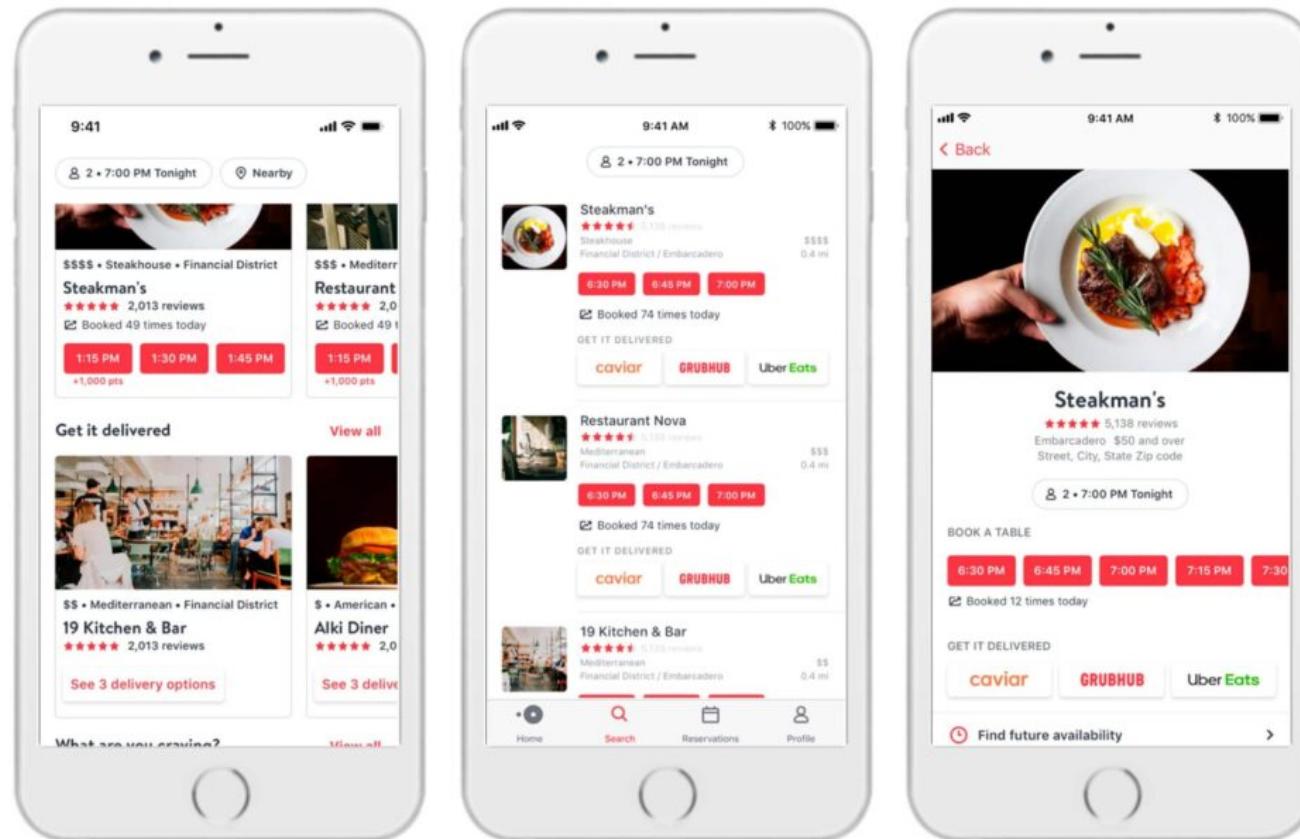
Конструювання програмного забезпечення



ПОШУК РЕСТОРАНІВ

Навіщо потрібен такий сервіс?

Щоб не їсти їжу з жовтих сумок.



ПОШУК РЕСТОРАНІВ

Функціональні вимоги:

- Користувачі можуть знаходити відповідні заклади поблизу.
- Користувачі можуть додавати/оновлювати/видаляти заклади.
- Користувачі можуть залишати відгуки до відвіданих місць із фото та оцінками

Нефункціональні вимоги:

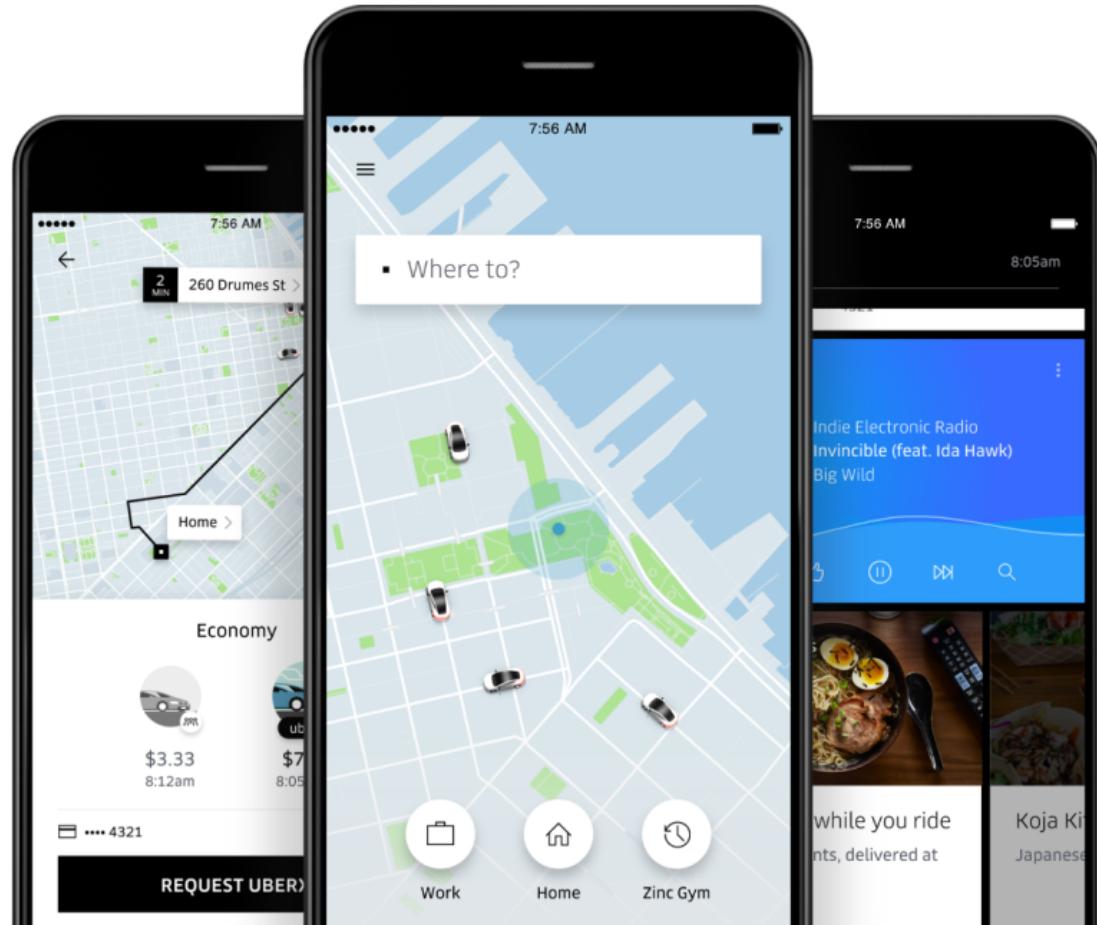
- Пошук має відбуватися швидко, інакше користувач замовить доставку.
- Набагато більше навантаження очікується на підсервіс пошуку, ніж на редагування.

Бонус: фільтри за категоріями, рекомендації нових закладів.

ПОШУК ТАКСІ

Навіщо потрібен такий сервіс?

Щоб не їздити на маршрутках.



Конструювання програмного забезпечення

ПОШУК ТАКСІ

Функціональні вимоги:

- Водії можуть виходити на зміну і чекати на замовлення.
- Пасажири можуть знаходити водіїв поблизу.
- Пасажири можуть замовляти поїздку, для неї знаходиться водій.
- Позиції відстежуються з моменту прийняття поїздки і до її закінчення. В кінці поїздки водій продовжує зміну.

Нефункціональні вимоги:

- Висока доступність сервісу - інакше пасажири пойдуть на метро, а водії на вокзал.
- Низький час очікування для пасажира, маленький простій для водія.

Бонус: оцінки водіям, оцінки пасажирам, пояснення ціноутворення.



РОЗРАХУНОК НАВАНТАЖЕННЯ НА СИСТЕМУ

Конструювання програмного забезпечення



ЗАГАЛЬНІ ВІДОМОСТІ

Після збору первинних **вимог** до системи дуже важливо оцінити необхідну **інфраструктуру** для розгортання і підтримки системи, а також **вартість** усього обладнання і процесів.

Від того, наскільки точно зробити розрахунок, залежатиме **успіх** проєкту і його **життєздатність**, а найголовніше - чи вистачить у замовника **матеріальних ресурсів**, щоб втілити всі сформульовані побажання і **вимоги** до системи.

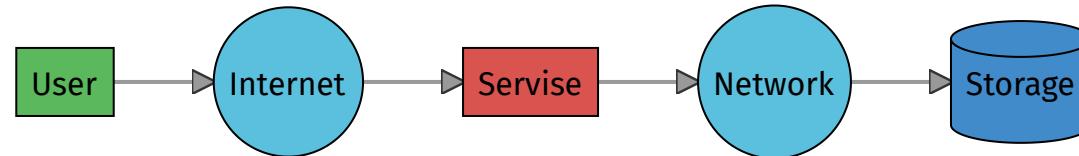
Оцінка дасть змогу скоригувати очікування від продукту.

РОЗРАХУНОК НАВАНТАЖЕННЯ НА СИСТЕМУ

На цій стадії проєктування ми оцінюємо з **різних сторін** яке навантаження може бути на систему.

Чи є якісь **перекоси** у бік **читання** або **запису** і яке **залізо** або **інтернет-канал** може знадобитися для передачі необхідної кількості даних.

З яких складових складається загальне навантаження?



ОСНОВНІ МЕТРИКИ РОЗРАХУНКУ

1. Трафік користувачів
2. Мережеве навантаження
3. Обчислювальне навантаження
4. Навантаження на сховище

ТРАФІК КОРИСТУВАЧІВ

Щоб розрахувати необхідний обсяг трафіку, потрібно оцінити **потенційну аудиторію** системи та підрахувати **загальну кількість користувачів**.

Якщо це якась **закрита система** для внутрішнього користування організації - кількість користувачів, виходячи з поточного **штатного розкладу**, обов'язків співробітників. Також можна закласти **відсоток розширення персоналу**, щоб врахувати приріст користувачів у майбутньому.

Якщо майбутня система це **загальнодоступний ресурс** - соціальна мережа або інший сервіс, тоді для приблизної оцінки можна проаналізувати **метрики найближчих конкурентів**.

Виходячи з кількості **потенційних користувачів**, далі можна оцінити час **роботи користувачів** у системі на день/місяць, оцінити **обсяг контенту**, що генерується і споживається під час роботи, а також обсяг **місця на жорстких дисках** під зберігання цього контенту.



ТРАФІК КОРИСТУВАЧІВ

Джентльменський набір користувачьких метрик:

1. Кількість активних користувачів в місяць (Monthly Active Users, MAU)
2. Кількість активних користувачів в день (Daily Active Users, DAU)
3. Загальна кількість користувачів на заданому горизонті (наприклад, 5 років)
4. Скільки та якого контенту у середньому створює один користувач на день

Від бізнес-показників до оцінки навантаження:

МЕРЕЖЕВЕ НАВАНТАЖЕННЯ

Мережеве навантаження складається з **утримуваних з'єднань** в одну одиницю часу і загального **обсягу трафіку**, який може пройти через мережу за одиницю часу.

Обмеження

1. **Кількість з'єднань:** C10k (1999), C1M (2011), C10M (2015).

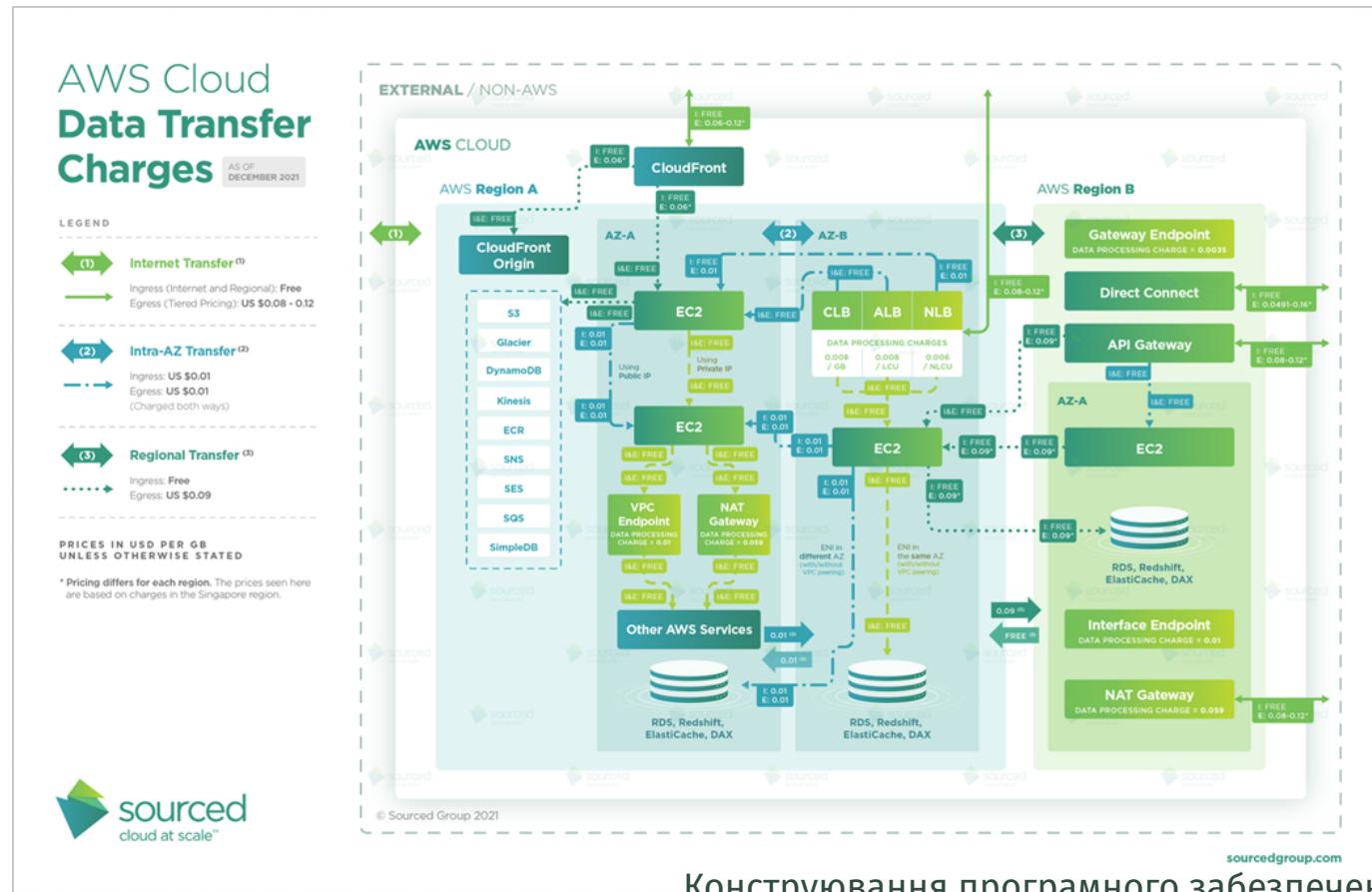
2. **Обсяг трафіку:**

- хмара – до 1GbE в секунду;
- вита пара – до 10GbE в секунду;
- оптоволокно стандарту QSFP+ – до 40GbE в секунду.
- Infiniband – до 100+ Gb в секунду.

МЕРЕЖЕВЕ НАВАНТАЖЕННЯ

З мережевого навантаження на сервіс можна вивести загальну вартість трафіку.

Хмарні провайдери, наприклад, оцінюють 1Gb трафіку в місяць в середньому від \$0.01 до \$0.1.



ОБЧИСЛЮВАЛЬНЕ НАВАНТАЖЕННЯ

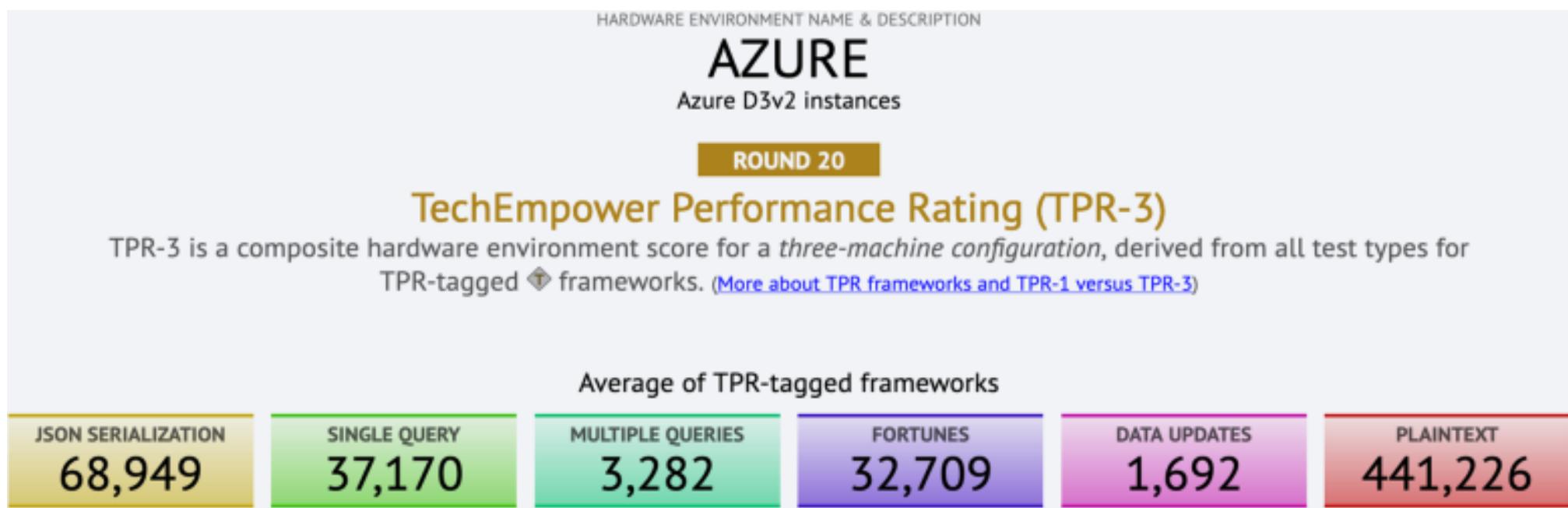
Під **обчислювальним навантаженням** зазвичай мається на увазі обчислення **кількості одночасних запитів** користувачів до сервісу. Цей показник обчислюється в **RPS** (requests per second).

Залежно від операції, з якою користувач звертається до сервісу, обчислювальне навантаження помітно різниеться, тому показники рахують для групи операцій, наприклад, окремо для операції на отримання текстових даних, окремо для читання з БД і окремо для запису в БД.

Обчислювальне навантаження на старті, звісно, оцінити найскладніше.

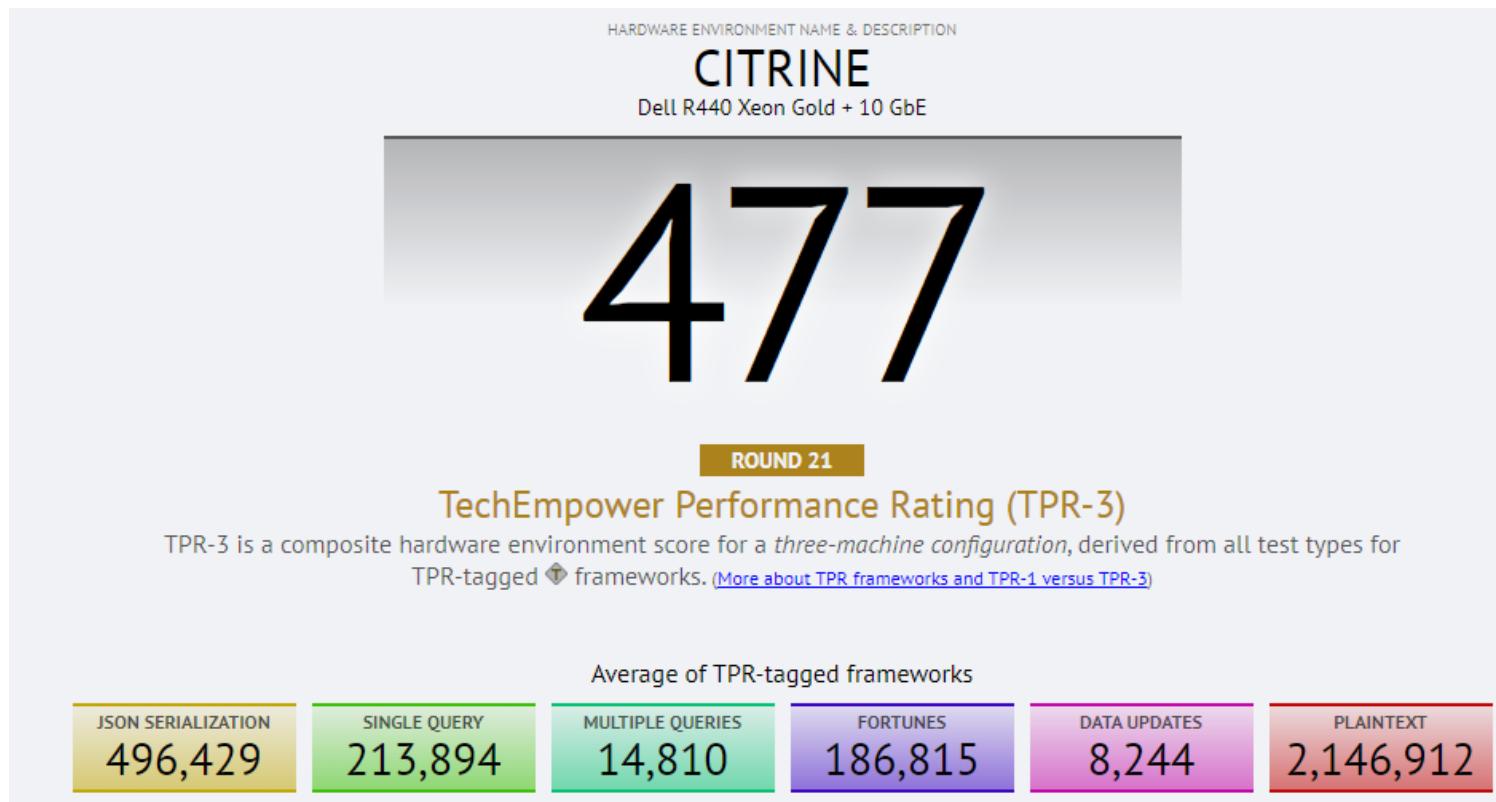
ОБЧИСЛЮВАЛЬНЕ НАВАНТАЖЕННЯ

Природним чином воно залежатиме від складності створюваної нами системи. Але на якийсь порядок цифр для простих сценаріїв можна орієнтуватися за бенчмарками TechEmpower Web Framework Benchmarks



За грубою оцінкою можна виходити з того, що в простих сценаріях у хмарі ми зможемо витримувати навантаження в 100k RPS для текстових даних і 10k RPS і 1k RPS під час читання і запису в БД.

ОБЧИСЛЮВАЛЬНЕ НАВАНТАЖЕННЯ



За наявності потужних **фізичних серверів** у тих самих сценаріях можна прикидати ймовірне навантаження в 500k RPS для текстових даних і 50k RPS і 5k RPS під час читання і запису в БД.

Реальні характеристики системи, звісно ж, виявлятимуться вже під час тестувань

НАВАНТАЖЕННЯ НА СХОВИЩЕ

Оцінивши кількість створюваного контенту з боку користувачів, ми можемо прикинути **скільки місця** він займатиме, який постійний **приріст** при цьому очікується і, відповідно, скільки того чи іншого заліза нам знадобиться для нашої системи.

Найпоширеніші види сховищ і значення показників їхньої швидкості обробки даних, а також вартість зберігання даних:

- **HDD** зі швидкістю 100-300 МБ/с, кожен диск зберігає до 20 ТБ за ціною до \$500.
- NVMe **SSD** зі швидкістю 3-5 ГБ/с, кожен диск зберігає до 8 ТБ за ціною до \$2000.
- Оперативна пам'ять (**RAM**) зі швидкістю 50 ГБ/с, вартість планки 128 ГБ близько \$1000.

НАВАНТАЖЕННЯ НА СХОВИЩЕ

Найпоширеніші види сховищ і значення показників їхньої швидкості обробки даних, а також вартість зберігання даних:

- **HDD** зі швидкістю 100-300 МБ/с, кожен диск зберігає до 20 ТБ за ціною до \$500.
- NVMe **SSD** зі швидкістю 3-5 ГБ/с, кожен диск зберігає до 8 ТБ за ціною до \$2000.
- Оперативна пам'ять (**RAM**) зі швидкістю 50 ГБ/с, вартість планки 128 ГБ близько \$1000.

У середньому зберігання **1 ТБ** обійдеться приблизно в \$10000 в **RAM**, \$300 на **SSD** і \$30 на **HDD**.

Можна прикидати, що в одному сервері буде до 1 ТБ **RAM**, 50 ТБ **SSD** або 200ТБ **HDD**.

Якщо дисків багато, у гру вступає і середня кількість відмов за рік (AFR) у розмірі 1%.

ЩЕ ПРО ПРОДУКТИВНІСТЬ

Один з важливих факторів для системи – це час/затримка відповіді або **latency**.

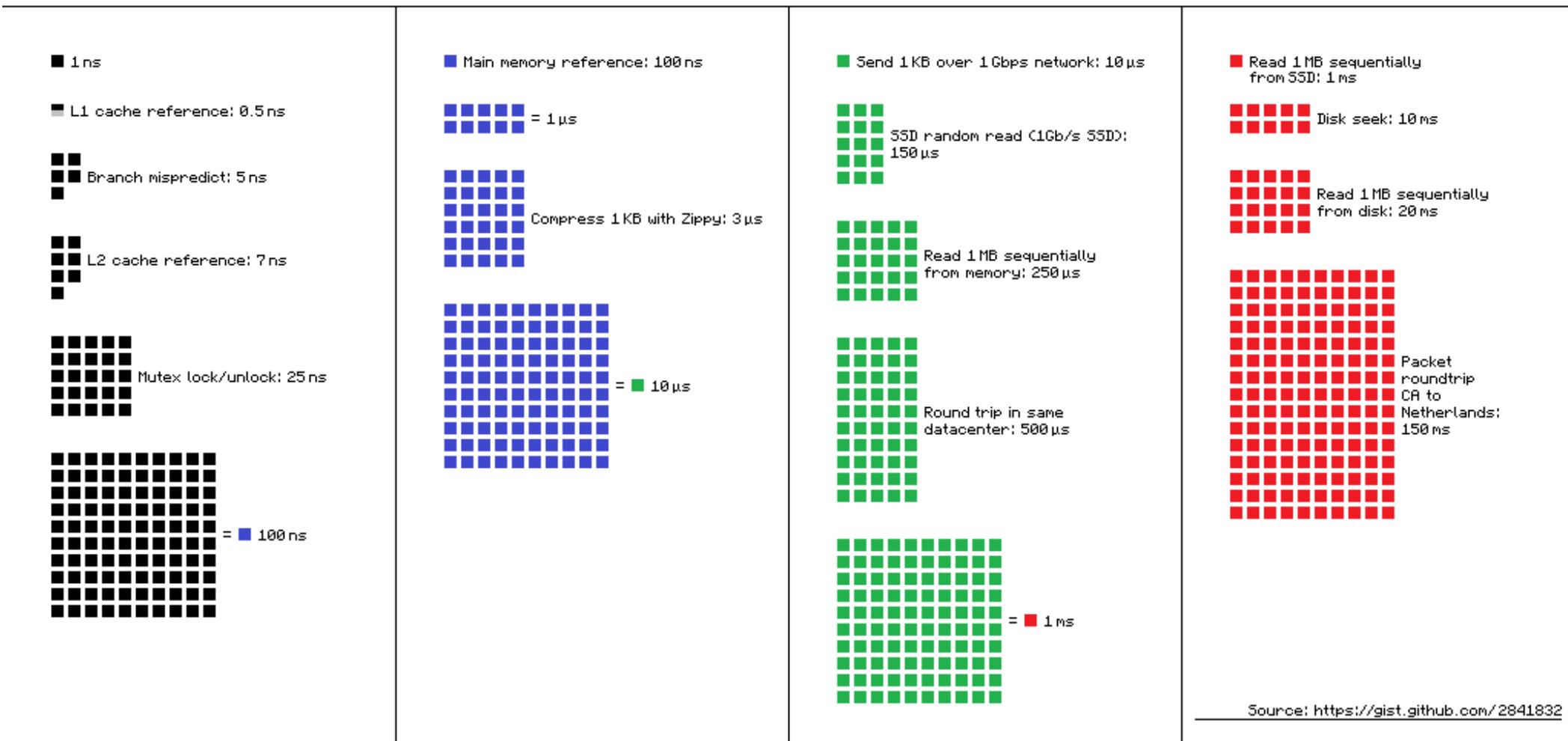
В деяких сценаріях вона буде визначатися швидкістю мережі/диску, але можуть бути й інші причини.

У 2011 році відомий інженер Jeff Dean з Google опублікував статтю зі своїми спостереженнями щодо часу виконання різних операцій на різних пристроях:

ЩЕ ПРО ПРОДУКТИВНІСТЬ

Те саме, але графічно:

Latency Numbers Every Programmer Should Know

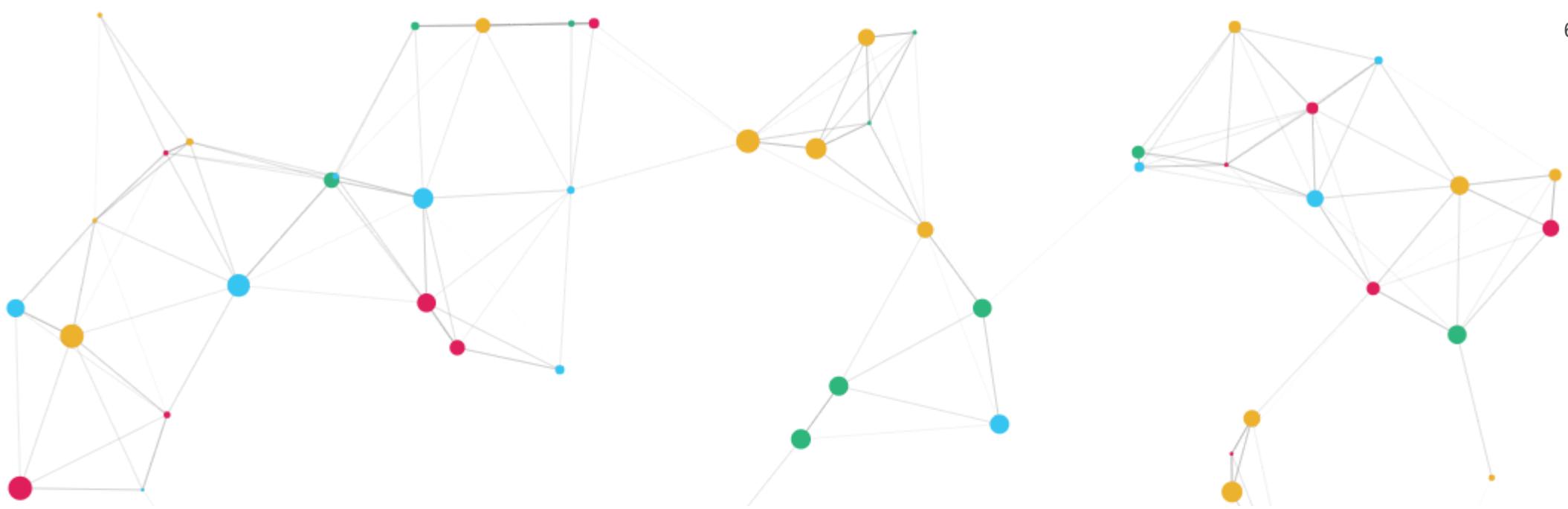


ЩЕ ПРО ПРОДУКТИВНІСТЬ

На які показники варто орієнтуватися під час розрахунків навантаження на систему:

- Трафік: **MAU, DAU**, приріст контенту на день, загалом за **5 років**, співвідношення **R/W**.
- Мережа: **C10k** давно не питання, **1 Gbps** навіть у хмарі, платимо не більше ніж **\$0.1/GB**.
- Обчислення: віддаємо **100к** тексту, зчитуємо **10к** і записуємо **1к** простих запитів у БД.
- Сховище: **HDD** 300 MB/s і \$30/TB, **SSD** 5 GB/s і \$300/TB, **RAM** 50GB/s і \$10000/TB.

В одну машину можна поставити **1 TB RAM, 50 TB SSD** або **200 TB HDD**.



ПРИКЛАДИ РОЗРАХУНКІВ



Конструювання програмного забезпечення

ПРИКЛАД 1: СЕРВІС КОРОТКИХ ПОСИЛАНЬ

Робимо оцінки виходячи з припущень:

- 100M скорочень щомісячно (MAU 10M, DAU 1M, 3 скорочення у середньому).
- Співвідношення читання до запису 100:1.

Звернень до сервісу в місяць: $100 * 100M = 10\text{Млрд.}$

Навантаження на створення записів: $100M / (30 * 86400) \approx 40 \text{ RPS.}$ На читання: $\times 100 = 4K \text{ RPS.}$

Якщо кожен запис займає 1 КБ, ми будемо генерувати трафік у 32 Мбит/сек¹.

На горизонті 5 років ми будемо зберігати $100M * 5 * 12 = 6\text{Млрд записів або 6 ТБ даних.}$

У підсумку, для таких користувачів достатньо і однієї робочої станції за декілька тисяч доларів.



ПРИКЛАД 2: ХОСТИНГ ТЕКСТІВ

Робимо оцінки виходячи з припущень:

- 1M завантажень в день (MAU 10M, DAU 500K, 2 завантаження у середньому).
- Співвідношення читання до запису 10:1.

Звернень до сервісу в місяць: $30 * 10 * 1M = 300$ млн.

Навантаження на створення записів: $1M / (86400) \approx 12$ RPS. На читання: $x10 = 120$ RPS.

Якщо кожен запис займає 10 КБ, ми будемо генерувати трафік у 10 Мбит/сек.

На горизонті 5 років ми будемо зберігати $30M * 5 * 12 = 1.8$ Млрд записів або 18 ТБ даних.

У підсумку, для таких користувачів достатньо і однієї робочої станції за декілька тисяч доларів.

ПРИКЛАД 3: АВТОДОПОВНЕННЯ

Робимо оцінки виходячи з припущень:

- 1 млрд. звернень до сервісу в день.
- 10 млн. унікальних запитів покриває більшу частину звернень.
- Запити складаються в середньому не більше ніж з 5 слів до 10 символів кожне.

Маємо: 50 символів на запит в середньому, тобто в 200 байт точно поміститься.

Всього на збереження всіх запитів за день піде: 10 млн. * 200 байт = 2 ГБ.

Якщо припустити, що кожен день з'являється 5% нових унікальних запитів, то на горизонті 5 років ми будемо зберігати індекс розміром $2 \text{ ГБ} * 365 * 5 * 1.05 = 3.8 \text{ ТБ}$.

У підсумку, для таких користувачів достатньо і однієї робочої станції за декілька тисяч доларів.

ПРИКЛАД 4: ХМАРНИЙ ДИСК

Робимо оцінки виходячи з припущень:

- 1 млрд. користувачів всього, 100 млн. DAU.
- У користувачів в середньому 5 гаджетів.
- У кожного користувача в середньому 100 файлів по 100 кБ, активні змінюють 1 файл.

Мережа: на оновлення 100 млн. * 100 кБ = 10 ТБ трафіку в день, 10 ТБ * 86 400 сек \approx 1 Gbps, 1 Gbps * \approx 2 тис. днів \approx 20 РВ за 5 років, відкрито 10 тис. з'єднань.

Обчислення: необхідно писати 100 млн. записів в день = 1 тис. RPS з записами до БД.

Сховище: всього зберігаємо 1 млрд. * 100 * 100 кБ = 10 РВ, тобто 50 серверів з 200 TB HDD кожен.

Вартість: 20 РВ трафіку обійтеться у \$2 млн., на зберігання 10 РВ HDD потрібно \$300 тис.

На датаноди нам знадобиться до 100 серверів, з'єднання і трафік витримає і один сервер, але краще більше.

Підтримка всій системи обійтеться в декілька млн. доларів на горизонті 5 років.

ПРИКЛАД 5: INSTAGRAM

Робимо оцінки виходячи з припущень:

- 1 млрд. користувачів всього, 100 млн. DAU.
- Користувач завантажує по одному фото в день розміром 100 кБ.
- Перегляди і завантаження співвідносяться 100:1.

Мережа: - завантаження фото 100 млн. / 100 тис. = 1 тис. RPS, 1 тис. * 100 кБ = 1 Gbps і \approx 200 PB за 5 років - читання 100 тис. RPS / 100 Gbps, C100k

Обчислення: 1 тис. RPS на запис фото, 100 тис. RPS на читання

Сховище: всього зберігаємо 20 PB фото, тобто 100 серверів з 200 TB HDD кожен.

Вартість: 200 PB трафіку обійтеться у \$2 млн., на зберігання 20 PB HDD потрібно \$600 тис.

На датаноди нам знадобиться до 100 серверів, з'єднання з трафіком та читання БД треба ще десятки.

Підтримка всій системи обійтеться в декілька млн. доларів на горизонті 5 років.

ПРИКЛАД 5: TELEGRAM

Робимо оцінки виходячи з припущень:

- 100 млн. DAU.
- Кожен користувач відправляє до 100 повідомлень в день, кожне 1 кБ.
- Кожне повідомлення читається 10 разів (приватні + групи).

Мережа: - завантаження повідомлень $100 \text{ млн.} * 100 / 86400 \approx 100 \text{ тис. RPS}$, 100 тис. * 1 кБ = 1 Gbps і $\approx 200 \text{ PB}$ за 5 років - читання 1 млн. RPS / 100 Gbps, C1M

Обчислення: 100 тис. RPS на запис повідомлень, 1 млн. RPS на читання

Сховище: всього зберігаємо 20 PB повідомлень, тобто 100 серверів з 200 TB HDD кожен.

Вартість: 200 PB трафіку обійтеться у \$2 млн., на зберігання 20 PB HDD потрібно \$600 тис.

На датаноди нам знадобиться сотні серверів і для датанод, і дня підтримки з'єднань, і для записів в БД.

Підтримка всій системи обійтеться в декілька млн. доларів на горизонті 5 років.

ПРИКЛАД 5: TWITTER

Робимо оцінки виходячи з припущень:

- 100 млн. DAU.
- Кожен користувач відправляє 1 повідомлення в день, кожне 100 кБ.
- Кожен користувач переглядає стрічку 10 користувачів.
- Читання та запис відбувається в пропорції 100:1.

Мережа: - завантаження повідомлень $100 \text{ млн.} * 1 / 86400 \approx 1 \text{ тис. RPS}$, 1 тис. * 100 кБ = 100 Mbps і $\approx 2 \text{ PB}$ за 5 років - читання 1 тис. RPS / 1 Gbps, C10k

Обчислення: 1 тис. RPS на запис повідомлень, 10 тис. RPS на читання + JOIN та алгоритмічна стрічка.

Сховище: всього зберігаємо 2 PB повідомлень, тобто 10 серверів з 200 TB HDD кожен.

Вартість: 2 PB трафіку обійтеться у \$200 тис., на зберігання 2 PB HDD потрібно \$60 тис.

На датаноди нам знадобиться десятки серверів + сервери для обчислень і допоміжні сервери.

Підтримка всій системи обійтеться в межах 1 млн. долларів на горизонті 5 років.

ПРИКЛАД 5: NETFLIX

Робимо оцінки виходячи з припущень:

- 10 млн. DAU, 20 000 фільмів у бібліотеці.
- Кожен користувач дивиться 1 годину контенту.
- Навантаження зі сторони контент-мейкерів поки пропускаємо.

Мережа: відео 1080р займає 5 MBps, трафік $10 \text{ млн.} * 4000 * 5 \text{ MBps} = 20 \text{ Pbps}$ ($20\ 000\ 000 \text{ Gbps}$ 😱).

Обчислення: $10 \text{ млн.} * 10 \text{ переглядів} / 100\ 000 \text{ сек.} = 1 \text{ тис. RPS}$ на отримання метаданих між серіями.

Сховище: 20 000 найменувань по 10 годин (фільми, але і серіали) $20 \text{ тис.} * 10 * 4000 * 10 \text{ Mbps} \approx 1 \text{ EB} = 1 \text{ млн. TB}$.

Вартість: На сховище необхідно витратити десятки мільйонів доларів, на мережу — **мільярди** 😱.

За трафік по стандартному прайсу розплатитися неможливо, тому Netflix має власні CDN (мережі дистрибуції контенту).

ДЯКУЮ ЗА УВАГУ!

✉ Матеріали курсу

👉 Data Mirosh

🗣 @aranaur

✉ ihor.miroshnychenko@kneu.ua

in @ihormiroshnychenko

🏡 aranaur.rbind.io

