

0. Scope refinement

Входит: работа с профилем, поиск по фильтрам, гео поиск, матчинг пар, чат

Не входит: платежная система, премиум подписки

1. Функциональные требования

1.1 личный кабинет пользователя (Profile UI)

- User service – добавление / редактирование / удаление профиля
- Upload service - загрузка / просмотр фото, видео

1.2 поиск анкет (Search UI)

- filter search service - поиск по фильтрам
- geo search service - поиск по геолокации
- wishlist service - возможность добавления анкет в избранное

1.3 матчинг пар (Matching UI)

- recoms service - рекомендация анкет
- like / dislike service - поставить лайк / дизлайк

1.4 общение в чате (Message UI)

- request service - выбрать человека и написать ему сообщение / принять сообщение
- chat service - управление перепиской

1.5 нотификация (Notification Provider) – внешний сервис

2. Нефункциональные требования

- общение в реальном времени с минимальными задержками
- консистентность в чатах важна
- консистентность в фото не критична
- высокая доступность
- высокая надежность

3. Оценка нагрузки

3.1 Метрики трафика

MAU = 100M

DAU = 10M

1 млн. ивентов / сек = 86 млрд. ивентов / сутки

Предположим:

- каждый пользователь отправляет по 50 сообщений в день

- каждый пользователь читает все отправленные им сообщения (чтение сообщений = запись сообщений)
- каждый пользователь загружает по 1 фото в день
- у каждого пользователя есть по 5 фото
- каждый пользователь просматривает по 10 анкет в день (чтение фото = $5 * 10 * \text{запись фото}$)
- каждый пользователь пользуется поиском 10 раз в день

3.2 Расчетные показатели по загрузке (load)

1 день = 10^5 секунд

3.2.1 Запросы на запись сообщений

$10\text{M (пользователи)} * 50 \text{ (сообщений / день)} = 10\text{M (пользователи)} * 50 \text{ (сообщений / } 10^5 \text{ секунд)} = 5000 \text{ (сообщений / с)} = 5000 \text{ RPS}$

3.2.2 Запросы на запись фото

$10\text{M (пользователи)} * 1 \text{ (фото / день)} = 10\text{M (пользователи)} * 1 \text{ (фото / } 10^5 \text{ секунд)} = 100 \text{ (фото / с)} = 100 \text{ RPS}$

3.2.3 Запросы на чтение сообщений

$10\text{M (пользователи)} * 50 \text{ (сообщений / день)} = 10\text{M (пользователи)} * 50 \text{ (сообщений / } 10^5 \text{ секунд)} = 5000 \text{ (сообщений / с)} = 5000 \text{ RPS}$

3.2.4 Запросы на чтение фото

$10\text{M (пользователи)} * 10 \text{ (анкет)} * 5 \text{ (фото / в анкете / день)} = 10\text{M (пользователи)} * 50 \text{ (фото / } 10^5 \text{ секунд)} = 5000 \text{ (фото / с)} = 5000 \text{ RPS}$

3.2.5 Запрос на поиск

$10\text{M (пользователи)} * 10 \text{ (поисков / день)} = 10\text{M (пользователи)} * 10 \text{ (поисков / } 10^5 \text{ секунд)} = 1000 \text{ (поисков / с)} = 1000 \text{ RPS}$

3.2.6 Запросы на ивенты (из условия)

$1\text{M (ивентов / с)} = 1\,000\,000 \text{ RPS}$

3.3 Сетевой трафик (network)

3.3.1 Сеть на запись сообщений (1 сообщение = 1 Кб)

$5000 \text{ (сообщений / с)} * 1 \text{ (Кб / сообщение)} = 5000 \text{ (Кб / с)} = 5 \text{ (Мб / с)} = 50 \text{ Mbps}$

3.3.2 Сеть на запись фото (1 фото = 100 Кб)

$100 \text{ (фото / с)} * 100 \text{ (Кб / фото)} = 10 \text{ (Мб / с)} = 0,1 \text{ Gbps}$

3.3.3 Сеть на чтение сообщений (1 сообщение = 1 Кб)

$5000 \text{ (сообщений / с)} * 1 \text{ (Кб / сообщение)} = 5000 \text{ (Кб / с)} = 5 \text{ (Мб / с)} = 50 \text{ Mbps}$

3.3.4 Сеть на чтение фото ($1 \text{ фото} = 100 \text{ KB}$)

$5000 (\text{фото} / \text{с}) * 100 (\text{Кб} / \text{фото}) = 0,5 (\text{Гб} / \text{с}) = 5 \text{ Gbps}$

3.3.5 Сеть на поиск ($1 \text{ поиск} = 10 \text{ Кб}$)

$1000 (\text{поисков} / \text{с}) * 10 (\text{Кб} / \text{поисков}) = 10 (\text{Мб} / \text{с}) = 0,1 \text{ Gbps}$

3.3.6 Сеть на ивенты ($1 \text{ ивент} = 1 \text{ Кб}$)

$1\text{М} (\text{ивентов} / \text{с}) * 1 (\text{Кб} / \text{ивент}) = 1 (\text{Гб} / \text{с}) = 10 \text{ Gbps}$

3.4 **Нагрузка на хранилище (storage)**

Расчет на 5 лет

3.4.1 профили пользователей (хранение фото)

$10\text{М} (\text{пользователи}) * 1 (\text{фото} / \text{день}) * 100 (\text{Кб} / \text{фото}) * 400 (\text{дней} / \text{год}) * 5 (\text{лет}) = 2 \text{ Пб}$

3.4.2 местоположение

4 байт – id, 8 байт на ширину и долготу

$10\text{М} (\text{пользователи}) * (4 + 8 + 8) \text{ байт} = 200 \text{ Мб}$

3.4.3 чаты (сообщения)

$10\text{М} (\text{пользователи}) * 50 (\text{сообщений} / \text{день}) * 1 (\text{Кб} / \text{сообщение}) * 400 (\text{дней} / \text{год}) * 5 (\text{лет}) = 1 \text{ Пб}$

3.4.4 ивенты

$1\text{М} (\text{ивентов} / \text{с}) * 1 (\text{Кб} / \text{ивент}) * 400 (\text{дней} / \text{год}) * 5 (\text{лет}) = 86 \text{ млрд} (\text{ивентов} / \text{сутки})$

$* 1 (\text{Кб} / \text{ивент}) * 400 (\text{дней} / \text{год}) * 5 (\text{лет}) = 2 * 10^{14} \text{ Кб} = 0,2\text{Еб}$

3.5 **Нагрузка на вычислительные мощности (hardware)**

3.5.1 Загрузка

Возьмем с запасом 1.5M RPS

1 инстанс = 10 000 RPS

У нас 1.5M RPS, значит необходимо минимум 150 инстансов

3.5.2 сетевой трафик

Возьмем с запасом 20 Gbps

1 инстанс = 1 Gbps

У нас 20 Gbps, значит необходимо минимум 20 инстансов

3.5.3 хранилище

Возьмем с запасом 0,3 Еб на 5 лет

1 SSD = 8 Тб

У нас 0,3 Еб, значит необходимо минимум 37500 SSD ~ 7500 инстансов

3.6 **Стоимость (cost)**

3.6.1 хранилище

Возьмем с запасом 0,3 Еб на 5 лет

Тариф - 1 ТБ = 300 \$ (SSD)

0,3 Еб * 300 (\$ / Тб) = 100 млн \$ на 5 лет (SSD)

Тариф - 1 ТБ = 30 \$ (HDD)

0,3 Еб * 30 (\$ / Тб) = 10 млн \$ на 5 лет (HDD)

3.6.2 сетевой трафик

Возьмем с запасом 20 Gbps = 2 Гб / с

2 Гб / с * 10⁵ (с / день) * 400 (дней / год) * 5 (лет) = 400 Пб трафика

Тариф - 1 Гб = 0,1 \$ (облако)

400 Пб * 0,1 \$ / Гб = 40 млн \$ на 5 лет

4. БД

Users – информация о юзерах (реляционная, MySQL)

Embeddings – информация о пользователях в векторном виде (векторная, Pinecone)

Matches – лайки / дизлайки (key-value, Redis)

Messages - Сообщения (документная, MongoDB)

Archive – хранение архивных переписок (документная, MongoDB)

Srorage – хранение фото / видео (хранилище S3)

Recoms – рекомендации (key-value, Redis)

Wishlists – сохраненные анкеты (реляционная, MySQL)

Cons – соединения между пользователями (key-value, Redis)

Cache – кешы ко всем бд кроме редиса (key-value, Redis)

QuadTree – квадродерево (кастомная реализация)

Positions – текущие локации пользователей (key-value, Redis)

5. Оптимизация

Для масштабируемости

- балансировщики между интерфейсами и основными бэкэнд сервисами
- гео шардирование - разделить данные по регионам
- добавление нескольких инстансов в основные сервисы
- в бд archive, recoms, messages добавлены кластера (шардирование + репликации)
- добавлено несколько инстансов бд QuadTree

Для повышения надежности

- Добавлена избыточность в балансировщики
- в бд users добавлены дополнительные реплики на чтение – много запросов на чтение будет
- в бд Cons, Positions, QuadTree, Matches добавлены офлайн копии (будут работать, если основные выйдут из строя)
- запасные сторонние провайдеры (для нотификации)

Для повышения отзывчивости

- в бд users добавлено кеширование (для активных или недавних пользователей)
- в бд messages добавлено кеширование (для недавних сообщений)
- Индексирование в бд users (чтение больше записей)

Для безопасности

- добавлен firewall

Чат – через WebSocket

ML –

Matching prioritizer (совместимость пользователей на основе их поведения и профилей),
anti-fraud (фейковые анкеты, мошеннические схемы,
NSFW-детектор (неприемлемые фото / сообщения).

Мониторинг – все сервисы присылают логи в очередь сообщений, затем они попадают в графит и в дашборд в графанае