

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СТРАТЕГИЙ КЭШИРОВАНИЯ В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ

Шуст И.В. (Университет ИТМО)

Научный руководитель – к.т.н Горлушкина Н. Н.
(Университет ИТМО)

Введение. В настоящее время далеко не во всех приложениях используют кэширование данных. Это связано со сложностью и стоимостью реализации, безопасностью пользовательских данных, а также малой эффективностью для некоторых видов приложений. В свою очередь это сказывается на производительности мобильных приложений. Так как без использования кэша приходится неоднократно “ходить” за данным в сеть, это в свою очередь негативно сказывается на пользовательском опыте. Все это ведет к понижению активности пользователей, и как следствие недополученным прибылям.

Основная часть. В большинстве мобильных приложений между долгими запросами в сеть используются различного рода тробберы, спиннер, скелетоны, шиммеры и много другое, что пользователь может распознать как некоего рода индикаторы выполнения операции, загрузки данных. Так пользователь понимает, что в скором времени он получит желаемый контент. Однако сеть может вести себя крайне непредсказуемо, пакеты могут теряться, токены доступа устаревать или просто пропадать доступ к сети. Все это ведет к ухудшению настроения пользователей, а как следствие к потере интереса и уходу из мобильного приложения. По мере накопления такого негативного опыта пользователь может оставить плохой отзыв о приложении в макрете, долгое время не заходить туда или вообще удалить продукт со своего мобильного устройства. В некоторых случаях наиболее худшим исходом может служить уход клиента к конкурентам, например в сфере банкинга.

Эту проблему можно решить несколькими путями:

- увеличить пропускную способность своих сервисов,
- кэширование.

Обычно выбирается первый путь, однако это ведет порой к излишнему усложнению структурной организации таких систем, к удорожанию ее масштабирования и обслуживания. Также стоит помнить, что повышение трафика “бьет” по финансам организации.

Однако необходимо понимать, что кэшировать все данные нельзя, так как часть из них может не иметь смысла и быть не актуальной. Кэширование подходит далеко не во всех ситуациях. Использование этого подхода может как и улучшить приложение, так и привести к уходу пользователей. Также стоит заметить, что изрядно большой кэш влечет за собой неочевидные последствия, связанные с внутренними ограничениями устройств, например приложение может аварийно завершать работу или просто не запускаться.

Различные стратегии кэширования отличаются по способу работы с данными и области их применения. Применимость, каждой из стратегии, определяется ее особенностями. Перед тем как выбрать стратегию кэширования для приложения, необходимо определиться необходима ли она вообще. Для ряда приложений кэширование данных является недопустимым или частично недопустимым. К таким приложениям можно отнести приложения из банковского сектора, приложения видеохостинги и другие. Данные, которые будут часто меняться или они каждый раз будут новыми, кэшировать не имеет смысла. Также в ряде случаев кэшировать персональные данные может быть не допустимым решением, тут все зависит от назначения приложения.

В одном приложении можно, а зачастую и необходимо, использовать несколько стратегий кэширования. У каждой стратегии есть свои плюсы и минусы, область применимости у них разная. Поэтому необходимо разбить весь функционал приложения на группы на основании сходства работы с данными. Для каждой из группы выбрать оптимальную стратегию кэширования. Также необходимо определить места хранения кэшируемых данных. Такими

местами могут быть: база данных, файл, оперативная память или preferences. При выборе места хранения стоит обращать внимание на скорость работы с хранилищем, его допустимый объем и сложность в реализации.

Реализовывать механизм кэширования лучше всего вынеся его в отдельный модуль с использованием интерфейса, чтобы реализация была скрыта. Сама реализация является оберткой над методами получения данных по сети. При вызове метода, получения данных, будет передаваться тип, на основании которого определяется к какой группе относятся данные. В зависимости от этого типа будет определяться стратегия кэширования.

Основной параметр, который характеризует стратегию кэширования – это процент попаданий запросов в кэш. Этот параметр довольно легко измерить, чтобы понять насколько выбранная стратегия кэширования эффективна.

Выводы. Использование кэша в мобильном приложении ведет к повышению скорости доставки контента пользователю. Быстрое отображение улучшает UX, что ведет к увеличению коэффициента удержания.

Сохранение данных в мобильном приложении ведет к уменьшению количества запросов в сеть, снижая трафик. А также ведет к повышению позитивного пользовательского опыта. Если правильно определиться со стратегией кэширования данных в конкретном модуле, то можно получить заметное повышение скорости предоставления услуг без потери качества.

Список использованных источников:

1. Стратегия кеширования в приложении [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/168725/> (дата обращения 08.02.2023)

2. Для чего нужно кеширование данных в мобильных приложениях [Электронный ресурс] — URL: <https://app72.ru/blog/288-dlya-chego-nuzhno-keshirovanie-dannykh-v-mobilnykh-prilozheniyakh> (дата обращения 08.02.2023)

3. Caching with Kotlin Coroutines [Электронный ресурс] — URL: <https://proandroiddev.com/caching-with-kotlin-coroutines-7d819276c820> (дата обращения 08.02.2023)

Шуст И.В. (автор)

Подпись

Горлушкина Н. Н. (научный руководитель)

Подпись