Оптимизация энергопотребления Android-приложения на основе применения «зелёных» паттернов

Тарбеев А.В., СПбГУ, Санкт-Петербург andrey@tarbeev.net,
Сабашный В.Е., Генеральный директор ООО «ЛАНИТ-ТЕРКОМ», Санкт-Петербург
vadim.sabashny@lanit-tercom.com

Аннотация

Энергоэффективность Android-приложений может достигаться путем использования специализированных оптимизаций со стороны разработчиков. В данной работе рассматривается вариант энергоэффективной оптимизации с использованием коллекции HashedMap из библиотеки Apache Commons Collections и переработкой паттерна декоратор.

Введение

В наше время мобильные устройства пользуются большой популярностью, и их число с каждым днем растет. С ростом популярности мобильных устройств выросли и требования к энергоэффективности аппаратных средств и программного обеспечения.

Важным пунктом на пути к выгодному использованию энергии является участие в процессе разработки всех, кто хотя бы как-то связан с мобильными устройствами. Инженеры создают энергоэффективные чипы и процессоры, а разработчики начинают пользоваться программными средствами для анализа потребления энергии приложением. Именно благодаря общим усилиям можно добиться максимальной энергоэффективности. Если, например, приложение будет написано неэффективно, то каким энергоэффективным не был бы процессор, устройство не сможет использовать запас заряда оптимально.

На помощь разработчикам программного обеспечения приходят не только инструменты профилирования, но и подходы в написании кода, получившие название «зелёных» паттернов, которые, как выясняется, тоже вносят свой вклад в энергопотребление. В последнее время ведутся работы по анализу влияния паттернов проектирования на потребление энергии.

В одной из работ [1] среди разработчиков был проведен опрос, чтобы выяснить их осведомленность о влиянии использования паттернов на энергопотребление. Результаты говорят о том, что большинство не учитывает их возможное влияние на энергозатраты при разработке. Исходя из результатов,

предположение о том, что open-source проекты содержат неэнергоэффективный код, является справедливым.

В данной работе предлагается провести обзор «зелёных» паттернов и далее внедрить набор оптимизационных решений, которые будут основано на использовании энергоэффективных структурах данных и на переработке паттерна декоратор, и оценить энергосберегающий эффект.

«Зелёные» паттерны

Понятие «зелёных» паттернов подразумевает под собой набор подходов, которые позволяют снизить энергопотребление приложения. Например, преимущественное использование темных оттенков для UI элементов, использование кэширования, выбор WiFi вместо мобильной сети. Их списки представлены в статьях [2, 3].

В статье [4] включено описание «зелёных» подходов, описания их влияния на энергопотребление. Например, вместо автоматического выполнения дейстия, являющегося энергозатратным, рекомендуется выполнять их по требованию пользователя. Или, например, заменить использование экрана в качестве источника информации на что-либо другое, например, на звуковые уведомления.

В работе [5] авторы сравнивали не одиночное применение «зелёных» подходов, а их совокупность. В результате они пришли к выводу, что некоторые комбинации могут негативно влиять на энергопотребление по сравнению с их отдельным применением.

В другой работе [6] авторы провели анализ влияния коллекций Java на энергопотребление. Они установили, что использование некоторых коллекций, например, $Hashmap$, ощутимо влияет на использование энергии. К тому же было установлено, что при смене коллекций на более энергоэффективные аналоги потребление снизилось сильнее у настольных систем в сравнении с мобильными системами. Дополнительно, был реализован инструмент, позволяющий давать рекомендации по замене коллекций.

В одной из работ [1] была затронута тема возможного влияния параметров, спрятанных внутри кода, на энергопотребление Android-приложений. Примеры таких параметров: размеры буфферов, размеры кэшей, количество потоков, частоты опросов и даже расположения UI элементов. Был реализован инструмент, который автоматически находил такие параметры, затем изменял их в надежде выяснить изменения в энергопотреблении. После каждого изменения авторы доступными средствами операционной системы снимали показания нагруженности аппаратных модулей, в последствии эти данные

использовались в энергетической модели, где и происходил расчет энергопотребления конкретного модуля. В результате проведенной работы авторы статьи пришли к выводу, что параметры практически не влияют на энергопотребление.

По результатам проведенного обзора «зелёных» паттернов выяснилось, что их использование в проектах с учетом их малой популярности среди разработчиков является набирающим популярность трендом.

Паттерны проектирования

Одной из значимых работ в области анализа влияния применения паттернов проектирования на энергопотребление является статья [7] C. Sahin et al. В ней авторы показали, что использование ряда паттернов проектирования может сказываться на энергопотреблении как в сторону увеличения, так и в сторону уменьшения. Так, по словами авторов, в $C++$ паттерн $Decorator$ увеличил энергопотребление на $\approx 700\%$, а паттерн $Flyweight$ уменьшил на $\approx 58\%$. В дополнение, паттерны, находящиеся по смыслу в одной категории ($creational$, $structural$, $behavioral$), не зависят от своей категории и могут как увеличивать, так и уменьшать энергопотребление.

В одной из работ [8] авторы провели анализ $Decorator$ паттерна в $Java$. В нем сравнивалось энергопотребление книжного примера кода кофейного автомата с паттерном и без него. Версия с рефакторингом, то есть без $Decorator$ паттерна, использовала на $\approx 96\%$ меньше энергии.

В работе [9] авторы оценили влияние применения ООП принципов, таких как наследование, полиморфизм, перегрузка операторов, на энергоэффективность. Наследование и полиморфизм не несут значительного влияния на энергопотребление и производительность, перегрузка операторов в свою очередь оказывает существенное влияние на энергоэффективность и производительность. Это связано с тем, что использование перегрузки может приводить к созданию временных объектов для промежуточных результатов, что и приводит к снижению производительности и увеличению энергопотребления.

Несмотря на то, что паттерны проектирования вносят структурированность и понятность в программную составляющую продукта, использование некоторых паттернов значительно влияет на энергозатраты, актуальной задачей остается их преобразование в более энергоэффективные аналоги.

По результатам проведенного обзора выяснилось, что внедрение «зелёных» паттернов и энергоэффективных паттернов проектирования является важным этапом в повышении энергоэффективности приложений, поэто-

му полученные результаты будут использоваться в процессе внедрения этих практик в open-source проект с дальнейшей оценкой энергопотребления.

Оптимизация

Описание экспериментов

В качестве тестовых Android-приложений были выбраны Remixed Dungeon[11] и Hentoid[12]. Они доступны публично, активно используются и развиваются.

Для проведения замеров были использованы смартфон (Vertex Luck L130, Cortex-A53, 2GB RAM), мультиметр (UNI-T UT803) и лабораторный генератор (QJE QJ3003H).

Смартфон, мультиметр и генератор были соединены в последовательную цепь. На генераторе фиксировалось напряжение, имеющее значение, которое совпадает со значением, указанным на аккумуляторе смартфона. Для минимизации влияния внешних факторов на замеры на смартфоне была выставлена минимальная яркость, отключены Wi-Fi и Bluetooth. Для фиксации тестового сценария использования приложений были написаны скрипты с использованием команд ADB (Android Debug Bridge). Время одного прогона составляло 15 минут. Также для устранения случайной ошибки замеров количество прогонов составляло десять повторений.

Использование энергоэффективных структур данных

Помимо стандартной библиотеки коллекций Java существуют сторонние библиотеки, например, Apache Commons Collections[10], которая содержит более оптимизированные в плане энергопотребления реализации коллекций. Нас будет интересовать сторонняя коллекция HashedMap из Apache Commons Collections. Она будет использоваться вместо стандартной реализации HashMap.

В Remixed Dungeon и Hentoid обычная Java-коллекция HashMap была заменена на HashedMap коллекцию из Apache Commons библиотки. По результатам замеров для двух приложений были получены результаты 0.5% и 0.6% соответственно.

Переработка паттерна декоратор

Существует несколько вариантов оптимизации паттерна декоратор. Создание объекта — это всегда ресурсоемкий процесс, заключающийся в поиске и аллокации необходимого пространства для новых объектов. Также из-за большого числа небольших создаваемых объектов растет нагрузка на сборщик мусора. В работе же будет рассмотрен вариант оптимизации с объединением всех декорируемых объектов для значительного сокращения числа создаваемых объектов.

В Remixed Dungeon приложении таким образом было сокращено число инстанциируемых сложных промежуточных объектов. В результате прирост составил 4.1%.

Вывод

Оптимизации понесли за собой ряд других проблем, которые необходимо принимать во внимание при решении об их внедрении.

Во-первых, это увеличение размера собранного приложения при использовании сторонней библиотеки для коллекций, Remixed Dungeon стало весить больше на 5 мегабайт, что не является критичным показателем в наше время. Однако для устройств, которые ограничены в количестве доступной памяти, это может стать важным фактором в решении о внедрении данной оптимизации.

Во-вторых, паттерны проектирования в первую очередь служат для того, чтобы написанный код был понятным и легкорасширяемым. Однако, стараясь угнаться за энергоэффективностью, мы теряем эти параметры. Стоит отметить, что известной практикой в игровой индустрии является использование анти-паттернов для повышения производительности в случае, если остальные способы оптимизации уже были внедрены, сюда же можно добавить и энергопотребление.

Заключение

Экспериментальным путем было показано, что влиять на энергопотребление Android-приложения возможно со стороны кода. Для приложений Remixed Dungeon и Hentoid после внедрения энергоэффективной коллекции HashedMap выигрыш в энергопотреблении составил 0.5% и 0.6% соответственно. Для приложения Remixed Dungeon после переработки паттерна декоратор выигрыш в энергопотреблении составил 4.1%.

Список литературы

[1] Xu, Qiang and Davis, James C. and Hu, Y. Charlie and Jindal, Abhilash. An Empirical Study on the Impact of Deep Parameters on Mobile App Energy Usage // 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). — С. 844–855. http://dx.doi.org/10.1109/SANER53432.2022.00103

[2] Nelson Gregório, João Paulo Fernandes, João Bispo, Sér- gio Medeiros. E-APK: Energy Pattern Detection in Decompiled Android Applications // Proceedings of the XXVI Brazilian Symposium on Programming Languages. — С. 50–58. http://dx.doi.org/10.1145/3561320.3561328

[3] Cruz Luis, Abreu Rui. Catalog of Energy Patterns for Mobile Applications // Empirical Softw. Engg. — С. 2209–2235. https://doi.org/10.1007/s10664-019-09682-0

[4] Daniel Feitosa and Luis Cruz and Rui Abreu and João Paulo Fernandes and Marco Couto and João de Sousa Saraiva. Patterns and Energy Consumption: Design, Implementation, Studies, and Stories // Software Sustainability. https://api.semanticscholar.org/CorpusID:240707094

[5] Couto, Marco and Saraiva, João and Fernandes, João Paulo. Energy Refactorings for Android in the Large and in the Wild // 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER). — С. 217–228. http://dx.doi.org/10.1109/SANER48275.2020.9054858

[6] Oliveira, Wellington and Oliveira, Renato and Castor, Fernando and Pinto, Gustavo and Fernandes, João Paulo. Improving Energy-Efficiency by Recommending Java Collections // Empirical Softw. Engg. — С. 3–45. https://doi.org/10.1007/s10664-021-09950-y

[7] Sahin, Cagri and Cayci, Furkan and Gutiérrez, Irene Lizeth Manotas and Clause, James and Kiamilev, Fouad and Pollock, Lori and Winbladh, Kristina. Initial explorations on design pattern energy usage // 2012 First International Workshop on Green and Sustainable Software (GREENS). — С. 55–61. http://dx.doi.org/10.1109/GREENS.2012.6224257

[8] Connolly Bree, Déaglán and Cinnéide, Mel Ó. Removing Decorator to Improve Energy Efficiency // 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). — С. 902–912. http://dx.doi.org/10.1109/SANER53432.2022.00108

[9] Maleki, Sepideh and Fu, Cuijiao and Banotra, Arun and Zong, Ziliang. Understanding the impact of object oriented programming and design patterns on energy efficiency // 2017 Eighth International Green and Sustainable Computing Conference (IGSC). — C. 1–6. http://dx.doi.org/10.1109/IGCC.2017.8323605

[10] Apache Commons Collections. https://commons.apache.org/proper/commons-collections/

[11] Remixed Dungeon https://github.com/NYRDS/remixed-dungeon

[12] Hentoid https://github.com/avluis/Hentoid