

УДК 004.45

ПРОГРАММНЫЙ СЕРВИС СБОРА ИНФОРМАЦИИ ОБ ЭНЕРГОПОТРЕБЛЕНИИ МОБИЛЬНОГО УСТРОЙСТВА

Гуляева С.А., Гришунов С.С., Белов Ю.С.

*Московский государственный технический университет имени Н.Э. Баумана, филиал,
Калуга, e-mail: fn1-kf@mail.ru*

В наши дни можно утверждать, что телефон перестал быть устройством только для звонков. Он позволяет нам оплачивать покупки, находить правильную дорогу, вызывать такси. Ситуация, в которой у вас садится элемент питания, становится одной из самых стрессовых. Остаться ночью на незнакомой улице без телефона довольно неприятно. При этом расход аккумуляторной батареи растет во многом как следствие расширения возможностей. Чтобы справиться с проблемой разряда батареи, необходимо сначала понять энергетическое поведение приложений и систем. Было разработано приложение, которое направлено на анализ информации о мощности устройства и событиях и на предоставление пользователям подробных данных о поведении энергии. Сервис состоит из серверной части, которая предоставляет информацию о мощности различных приложений, и клиентской стороны, которая анализирует информацию о мощности и событиях для конкретных приложений. В статье оценивается работа данного сервиса, позволяющего собирать данные об энергопотреблении устройства, на планшете и смартфоне. Предоставляются данные о том, как работает сервис для трех групп пользователей: сценарий конечного пользователя, сценарий разработчика приложения, сценарий разработчика системы. Также описаны подходы к тестированию и протестирован разработанный сервис.

Ключевые слова: энергоэффективность устройств, тестирование, анализ, клиент-серверное приложение, подходы к тестированию

SOFTWARE SERVICE CARRYING OUT THE COLLECTION OF INFORMATION ON THE POWER CONSUMPTION OF THE MOBILE DEVICE

Gulyaeva S.A., Grishunov S.S., Belov Yu.S.

Bauman Moscow State Technical University, Kaluga branch, Kaluga, e-mail: fn1-kf@mail.ru

Nowadays it can be argued that the telephone is no longer a device for making calls. It allows us to pay for purchases, find the right road, call a taxi. The situation in which your battery is low becomes one of the most stressful. Staying on an unfamiliar street at night without a phone is rather unpleasant. At the same time, battery consumption is increasing largely as a result of expanding capabilities. To deal with the problem of battery drain, you must first understand the energy behavior of applications and systems. An application has been developed that aims to analyze information about device power and events and to provide users with detailed data on energy behavior. The service consists of a server side, which provides information about the capacity of various applications, and a client side, which analyzes information about capacity and events for specific applications. This paper evaluates the work of this service, which allows you to collect data on the energy consumption of a device on a tablet and a smartphone. It provides data on how the service works for three groups of users: end-user script, application developer script, system developer script. Approaches to testing are also described, and the developed service is tested.

Keywords: energy efficiency of devices, testing, analysis, client-server application, approaches to testing

В настоящее время мобильные устройства, такие как планшеты и смартфоны, стали важной частью нашей повседневной жизни. Нет никаких сомнений, что приложения делают нашу жизнь более удобной и красочной, но не стоит забывать, что они потребляют большое количество энергии на мобильных устройствах и значительно влияют на время автономной работы устройства. Конечно, пользователю необходимо знать, какое приложение является более энергозатратным при одном и том же наборе функций. Энергоэффективные приложения более конкурентоспособны на рынке.

Цель исследования: провести тестирование и анализ разработанного приложения.

В ходе работы был разработан сервис, который представляет собой профилировщик и анализатор мощности прикладно-

го уровня. Как показано на рис. 1, сервис возвращает конечным пользователям информацию о мощности соответствующих приложений и дает им дополнительные рекомендации при выборе интересующих их приложений. Для разработчиков приложений, помимо информации о мощности аналогичных приложений, собранной с сервера, клиент также показывает информацию о событиях их разработок, поэтому можно легко найти проблемы с питанием приложений. С точки зрения разработчиков систем, детальная информация о мощности системы, предоставляемая клиентом, полезна им для выбора механизмов энергосбережения [1].

Описание подходов к тестированию. Сервис получает системную информацию от двух классов, PowerProfile и BatteryStats,

и вычисляет энергопотребление приложения на основе моделей мощности. Чтобы проверить точность считывания данных из класса PowerProfile, сначала сравнивается файл ресурсов между различными версиями ОС Android. Кроме того, были написаны собственные тесты, чтобы проверить, согласуются ли результаты с данными, записанными в файле. В основном тестировались яркость, процессор, подключение сокета и файловый ввод / вывод. В экспериментах эталонные приложения запускались на переднем плане, в то время как работа остальных приложений была остановлена для сохранения точности расчетов. Исходя из полученных данных, можно рассчитать мощность, а также информацию о потребляемой энергии. Файл ресурсов содержит данные о мощности различных состояний экрана, Wi-Fi, процессора, bluetooth и так далее. Были запущены тесты, чтобы сравнить собранные данные с информацией, содержащейся в файле. Кроме того, произведен расчет данных ввода-вывода файла, а результаты помещены в систему расчета мощности, которая не поставляется в файле ресурсов.

Мобильные устройства, которые использовались для этих экспериментов – это Huawei Mate 10 Pro и Samsung Galaxy Tab A 8.0. Их аппаратные параметры представлены в табл. 1, включающей информацию о датчиках. Оба этих устройства используют ОС Android 8.0. Чтобы обеспечить репрезентативные результаты, выбранные приложения охватывают большинство категорий. Они имеют более миллиона установок и входят в топ-100, согласно Google Play.

Тестирование сервиса. Были сделаны три тематических исследования, чтобы проиллюстрировать, как различные типы пользователей могут воспользоваться преимуществами сервиса.

Сценарий конечного пользователя. Конечный пользователь обычно хочет, чтобы батарея работала дольше, не слишком часто заряжаясь. Помимо экономии энергии с помощью операционной системы или отключения неиспользуемых устройств, эта цель может быть достигнута путем установки энергосберегающих приложений. Разработанный сервер поддерживает большое количество данных о мощности приложений с информацией об аппаратной платформе.

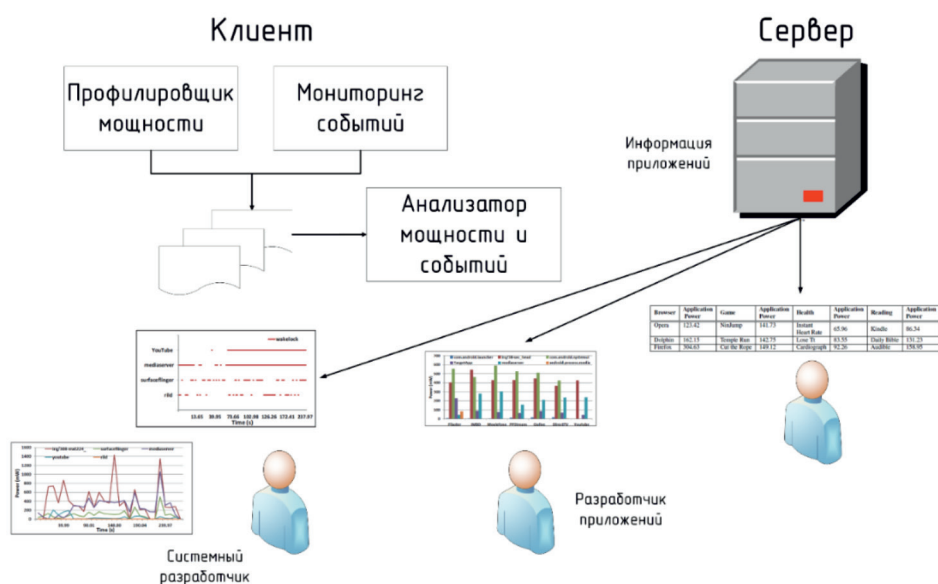


Рис. 1. Схема работы сервиса

Таблица 1

Экспериментальные платформы

Компонент	Huawei Mate 10 Pro	Samsung Galaxy Tab A 8.0
Процессор	8 ядер, 4 x A73 (2,36 ГГц) + 4 x A53 (1,8 ГГц)	Qualcomm Snapdragon 429
Частота (ГГц)	2	2
RAM (Гб)	4	2

Конечные пользователи могут запрашивать эту информацию и искать категорию приложения, которое они хотят установить, возвращаемые данные ранжируются по энергопотреблению приложений. Кроме характеристик приложения, таких как пользовательский интерфейс и специальные функциональные возможности, улучшающие пользовательский опыт, конечные пользователи также могут учитывать энергопотребление [2].

Возьмем в качестве примера браузер, предположим, что пользователи хотят установить Opera на свое устройство. Они могут просто отправить тип устройства и имя приложения Opera на сервер. В табл. 2 перечислены данные нескольких приложений, хранящиеся на сервере.

Отмечено, что данные по мощности рассчитываются в соответствии с общей ситуацией использования. В этом эксперименте были выбраны шесть популярных веб-сайтов, включая YouTube.com, Wikipedia.org., Amazon.com, открыли их один за другим и каждый раз прокручивали вниз, чтобы увидеть всю информацию [3]. Можно увидеть, что Opera потребляет меньше энергии, чем Firefox, что делает ее более конкурентоспособной. Чтобы выяснить причины этого, необходимо проанализировать информацию о событиях и необработанные данные журнала питания Opera и Firefox, результаты показывают, что их мощность процессора имеет большую разницу. Firefox может делать больше обработки и вычислений для улучшения пользовательского опыта.

Сценарий разработчика приложений. С одной стороны, сервер предоставляет данные о мощности приложений, которые впоследствии используются разработчиками приложений. С другой стороны, разработчики могут получать информацию о событиях от клиента, что дает направление оптимизации с точки зрения энергопотребления. В этом разделе используется видеоприложение в качестве примера, чтобы показать, как работает сервис [4].

Чтобы учесть влияние, которое приложение может оказать на систему, сервер не только предоставляет информацию о каждом приложении, но и дает другие четыре наиболее энергозатратных процесса каждого приложения и сравнивает их объединение. Таким образом, на рис. 2 сравниваются шесть процессов. Данные собираются из Huawei Mate 10 Pro, и можно увидеть, что системные процессы, поддерживающие приложения, потребляют гораздо больше энергии, чем само приложение. Например, systemui отвечает за рисование пользовательского интерфейса, mediaserver обеспечивает звуковую и другую поддержку мультимедиа. С точки зрения целевого приложения YouTube находится в хорошей ситуации, его мощность ниже, чем у других. На рис. 3 представлена информация о событиях и изменении мощности приложения YouTube как на Huawei Mate 10 Pro, так и на Samsung Galaxy Tab A 8.0. Эта информация помогает разработчикам понять проблему питания. Для информации о событиях ось x – это время; ось y – это процессы, которые генерируют эти события [5]. Записанные события включают в себя wakelock (блокировка сна), датчик, экран и т.д. Была найдена только информация wakelock в этом сценарии, процесс mediaserver появился в обоих устройствах, а rild, surface flinger занимают wakelock на Huawei Mate 10 Pro, а systemui на стороне Samsung Galaxy Tab A 8.0. YouTube также занимает wakelock в течение длительного времени, как показано на рис. 3, а и в, разработчики могут анализировать свой код для улучшения использования wakelock, например, выпускать wakelock в состоянии паузы приложения. Изменение мощности процессов показано на рис. 3, б и д, которые занимают событие или имеют высокое энергопотребление, YouTube потребляет высокую мощность, когда запускается приложение, в то время как mediaserver периодически достигает высшей точки.

Таблица 2

Сравнение мощности различных приложений

Браузер	Мощность приложения (мВт)	Игра	Мощность приложения (мВт)
Opera	123.42	NinJump	141.73
Dolphin	162.15	Temple Run	142.75
Firefox	304.63	Cut the Rope	149.12
Здоровье	Мощность приложения (мВт)	Чтение	Мощность приложения (мВт)
Instant Heart Rate	65.96	Kindle	86.34
Lose Tr	83.55	Daily Bible	131.23
Cardiograph	92.26	Audible	158.95

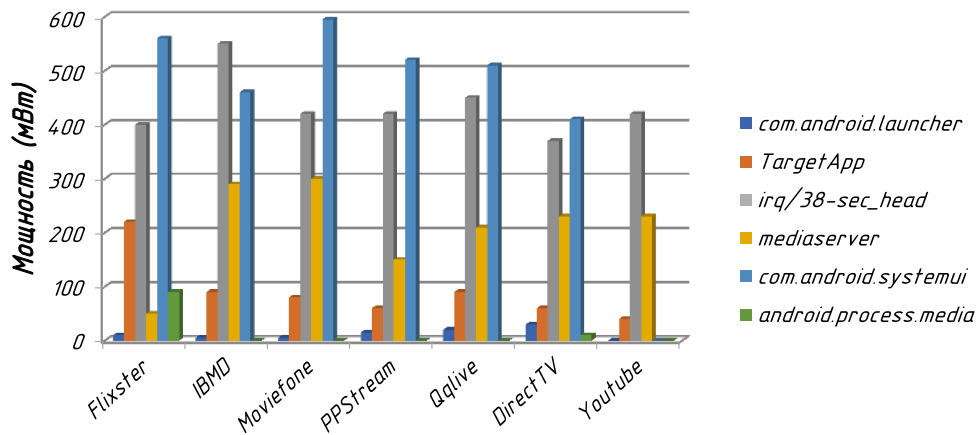
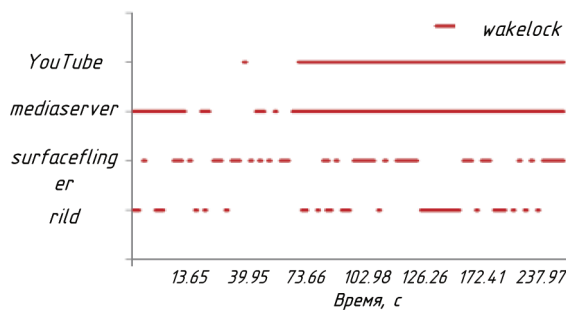
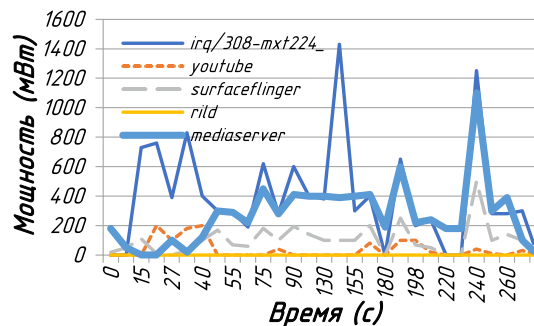


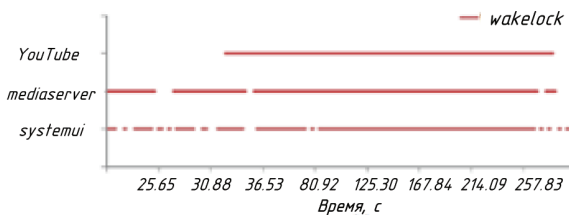
Рис. 2. Сравнение мощности семи видеоприложений



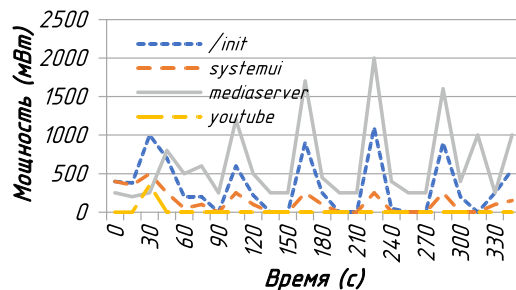
(а) информация о событиях YouTube, запущенного на Huawei Mate 10 Pro



(б) Информация об изменении мощности на Huawei Mate 10 Pro



(в) информация о событиях YouTube, запущенного на Samsung Galaxy Tab A 8.0



(д) Информация об изменении мощности на Samsung Galaxy Tab A 8.0

Рис. 3. Сравнение событий YouTube и информации о мощности

Проанализировав информацию об использовании ресурса в начале YouTube, было обнаружено, что он передавал сетевые пакеты и имел дело с пользовательскими входами (например, touch, click). Никаких других лишних данных не обнаружено. Еще одной причиной столь высокого энергопотребления является система подготовки к началу новой деятельности.

Сценарий разработчика системы. Разработчики систем больше заботятся о потребле-

нии энергии всей системой, а не о конкретном приложении или аппаратном компоненте. Сервис предоставляет информацию о мощности всех процессов, запущенных в системе, что точно помогает знать всю информацию о потребляемой энергии. Раньше было выяснено, что системные процессы потребляют гораздо больше энергии, чем само целевое приложение. Чтобы показать, что это обычная проблема, был проведен еще один эксперимент с игровыми приложениями [6].

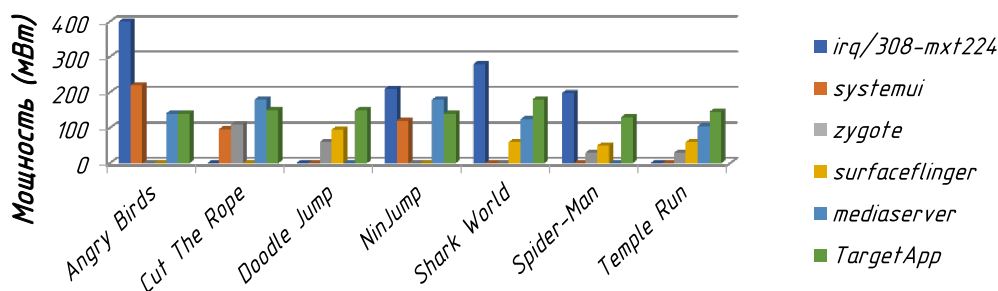
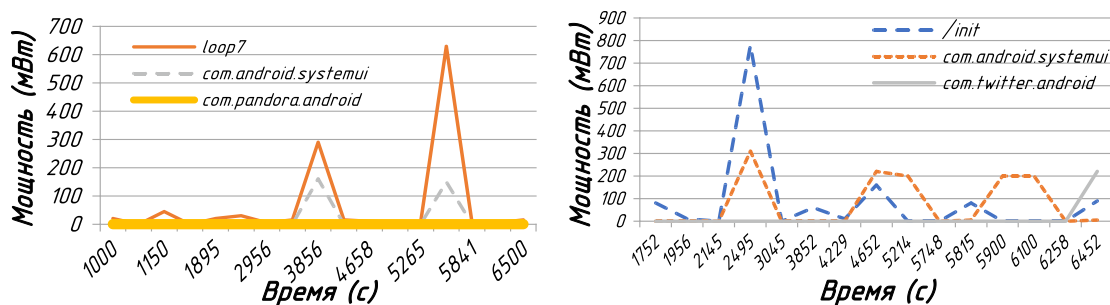


Рис. 4. Сравнение мощности семи игровых приложений



(а) Изменение мощности на Huawei Mate 10 Pro (б) Изменение мощности на Samsung Galaxy Tab A 8.0

Рис. 5. Сравнение информации о питании устройств в режиме «сна» с фоновыми приложениями

Были проанализированы 7 популярных игр: Angry Birds, Cut the Rope, NinJump и так далее. Энергопотребление каждой игры и нескольких соответствующих системных процессов на Huawei Mate 10 Pro демонстрируется на рис. 4. Можно наблюдать, что системные процессы, такие как irq/308-mxt224, mediaserver и zygote, потребляли много энергии, следовательно, они не были незначительными по сравнению с целевыми приложениями.

Кроме активных приложений, фоновые приложения также являются основной проблемой для разработчиков систем. В экспериментах сначала проверялась ситуация, когда существуют только системные процессы и не открываются никакие другие приложения. Чтобы представить реальный случай, когда пользователи используют эти устройства, были проведены эксперименты с приложениями, работающими в фоновом режиме. Наиболее распространенная ситуация, которую он представляет, это переход пользователей в спящий режим, их мобильные устройства находятся в спящем состоянии, не выходя из всех открытых приложений [7]. На рис. 5 показано изменение мощности трех основных энергоемких процессов Huawei Mate 10 Pro и Samsung

Galaxy Tab A 8.0 в режиме «сна» с незамкнутыми приложениями. Прежде чем перевести устройства в режим «сна», были открыты facebook, twitter, youtube, angrybirds и pandora и каждое использовалось в течение нескольких минут. Из рисунка можно узнать, что мощность большинства приложений невелика, в то время как системные процессы все еще потребляют достаточное количество энергии. Поэтому разработчики систем должны уделять больше внимания оптимизации этих фоновых процессов и служб.

Заключение

Сервис состоит из серверной части, которая предоставляет информацию о мощности различных приложений, и клиентской стороны, которая анализирует информацию о мощности и событиях для конкретных приложений. Были показаны примеры поиска первопричин большого энергопотребления, например чрезмерное использование wakelock [8]. Анализ информации о мощности приложений полезен для многих исследований, связанных с энергией/мощностью на мобильных устройствах, и выводы, полученные из наблюдений, указывают на несколько потенциальных направлений оптимизации.

Список литературы

1. Гуляева С.А., Белов Ю.С. Подходы к оптимизации энергопотребления программно-аппаратных компонентов смартфона // E-Scio: Электронное периодическое издание «E-Scio.ru». 2020. № 7. С. 520–529.
2. Гуляева С.А., Ефимов П.В., Белов Ю.С. Влияние уровней архитектуры мобильных устройств на общее потребление энергии // E-Scio: Электронное периодическое издание «E-Scio.ru». 2019. № 12. С. 669–674.
3. Park S., Kim D., Cha H. Reducing energy consumption of alarm-induced wake-ups on android smartphones. Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. 2015. P. 33–38.
4. Benkhelifa E., Welsh T., Tawalbeh L., Jararweh Y. User Profiling for Energy Optimisation in Mobile Cloud Computing. Procedia Computer Science. 2015. No. 52. P. 1159–1165.
5. Tawalbeh L., Basalamah A., Mehmood R., Tawalbeh H. Greener and Smarter Phones for Future Cities: Characterizing the Impact of GPS Signal Strength on Power Consumption. IEEE Access. 2016. No. 4. P. 858–868.
6. Qian H., Andresen D. Extending mobile device's battery life by offloading computation to cloud. Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems. 2015. P. 150–151.
7. Bolla R., Khan R., Parra X., Repetto M. Improving smartphones battery life by reducing energy waste of background applications. Next Generation Mobile Apps, Services and Technologies (NGMAST), Eighth International Conference. 2014. P. 123–130.
8. Sewook Park, Dongwon Kim, Hojung Cha. Reducing energy consumption of alarm-induced wake-ups on android smartphones. Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. 2015. P. 33–38.