

УДК 004.55

Е. В. СОКОЛОВА

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ В БИЗНЕС-ПРИЛОЖЕНИЯХ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID

Рассмотрено использование моделей сред реализации облачных вычислений в бизнес-приложениях для мобильных устройств. Обоснован выбор облачной серверной инфраструктуры BaaS для мобильных приложений. Определены особенности разработки программного обеспечения для мобильных устройств с использованием облачных вычислений. Описан эксперимент для проверки эффективности использования модели реализации среды облачных вычислений BaaS. Проведен анализ экспериментально полученных результатов, выявлены негативные и позитивные стороны использования облачных вычислений в разработке бизнес-приложений для мобильных устройств на базе ОС Android.

Ключевые слова: программное обеспечение, облачные вычисления, мобильные устройства, метрики исходного кода.

Введение

С началом информационной эпохи стала возрастать потребность общества обрабатывать большие объёмы информации и предоставлять доступ к данным в произвольный момент времени. В связи с этим в ИТ-индустрии интенсивно развивается отрасль разработки приложений для мобильных устройств [1]. Возможность доступа к любому контенту, на любом устройстве в любой сети независимо от географического местоположения, становится реальностью. По данным исследований выручка на мировом рынке мобильных приложений в 2014 г. составила \$33,7 млрд, что на \$12,3 млрд больше показателя 2013 г., прогнозируется, что в 2018 г. объём рынка преодолет отметку в \$60 млрд, то есть среднегодовой темп роста составляет 15,5 % [2]. В организациях рассматривают вопросы о расширении числа работников, использующих в бизнес-процессах смартфоны. Основными проблемами, с которыми сталкиваются разработчики приложений для мобильных устройств, являются: большое количество различных типов мобильных устройств и платформ, огромные объёмы и разнообразные типы хранящихся данных, а также возрастающая угроза безопасности [3].

Для устранения этих проблем в настоящее время разработчиками программного обеспечения (ПО) используется подход, обеспечивающий хранение информации с использованием сред облачных вычислений (ОВ). Среда ОВ - это программно-аппаратная модель средств вычислительной техники, позволяющая получать удалённый доступ к вычислительным ресурсам в любой момент времени, позволяет динамически выделять требуемое для ПО

процессорное время и память в зависимости от текущей нагрузки на это ПО. При этом доступ к ПО, выполняемому в среде ОВ, обеспечивается посредством сети Интернет.

1. Анализ публикаций

Среды ОВ являются следующим звеном в эволюционной цепочке моделей предоставления удалённого доступа к данным после модели выделенных серверов. Выделенный сервер это - программно-аппаратная платформа, обеспечивающая доступность информации, её хранение и защиту. Однако, в случае отключения выделенного сервера от сети Интернет или повышения на него нагрузки, программное обеспечение, выполняемое на выделенном сервере, становится недоступно пользователям. Кроме того, недостатком выделенных серверов является необходимость резервирования значительного количества ресурсов для обеспечения доступности ПО при пиковых нагрузках или случаях отказа аппаратной составляющей. При этом ПО выделяется большее количество ресурсов, чем ему необходимо, что приводит к их неоптимальному использованию.

С увеличением пропускной способности каналов передачи данных, предложенная концепция стала осуществимой, и поставщики услуг доступа к данным начали использовать среды облачных вычислений, которые лишены вышеперечисленных недостатков выделенных серверов. Впервые услуга предоставления доступа к данным на основе среды ОВ была предложена компанией Salesforce в 1999 году [4]. Впоследствии, услуги предоставления доступа к данным с использованием сред ОВ стали предлагать на рынке Amazon, Google, Microsoft и

множество других компаний, а использование облачных вычислений стало общей практикой. Появляются платформы Backend as a Service (BaaS), которые предоставляют готовую облачную серверную инфраструктуру для всех типов приложений, что позволяет разработчикам, стартапам и крупным компаниям выигрывать время и деньги, отказавшись от разработки своего сервера, и сфокусироваться на функциональности приложений, их продвижении и улучшении интерфейса пользователя.

Актуальность темы исследования определяется внедрением моделей ОБ в практические решения разработчиков бизнес-приложений для мобильных устройств.

Основная **цель работы** – на основе экспериментальной оценки метрик сложности ПО экспериментально обосновать целесообразность использования модели реализации среды облачных вычислений BaaS в разработке мобильных приложений.

Метрика программного обеспечения – это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. Метрики необходимы для упорядочения программных продуктов по выбранным свойствам, которые они характеризуют [5].

Среди всех метрик стадии разработки ПО особый интерес представляют метрики размера и сложности ПО, так как они лучше прочих поддаются формальной оценке, а сложность является основополагающим фактором качества ПО, так как чем ниже сложность, тем проще программисту ориентироваться в программе и писать новый код, ниже вероятность внести ошибку и выше шанс обнаружить уже существующую.

Несмотря на преимущества метрического подхода к оценке качества ПО, формальные метрики сами по себе не могут быть использованы в качестве основы для вынесения решения о качестве продукта [6]. Существует множество параметров, которые не могут быть оценены формально, например, производительность труда программиста или совершенство архитектуры ПО. С другой стороны, метрики могут быть использованы в качестве хорошего дополнения к прочим методам оценки качества ПО.

Для достижения этой цели необходимо решить следующие **задачи**:

- провести анализ проблем, связанных с использованием моделей сред реализации облачных вычислений в бизнес-приложениях для мобильных устройств;

- обосновать выбор модели реализации среды облачных вычислений в бизнес-приложениях для мобильных устройств;

- спланировать и провести эксперимент для проверки эффективности использования модели

реализации среды облачных вычислений BaaS;

- проанализировать полученные результаты эксперимента и выявить негативные и позитивные стороны использования облачных вычислений в разработке бизнес-приложений для мобильных устройств на базе ОС Android.

2. Особенности разработки ПО для мобильных устройств с использованием ОБ

Основные принципы проектирования ПО для мобильных устройств неизменны независимо от платформы и операционной системы, установленной на ней. Главными отличиями являются технические особенности, такие как инструментарий, API и SDK, парадигмы проектирования интерфейса [7].

Несмотря на ежегодное увеличение производительности смартфонов в кратном размере, эти устройства по-прежнему имеют ряд ограничений на использование оперативной памяти, уровень нагрузки графическими материалами, быстроту обработки информации, ограниченный срок работы устройства от аккумуляторной батареи. В связи с этим программист обязан внимательно следить за выделением памяти и своевременным удалением неиспользуемых объектов. Технологии облачных вычислений и мобильных устройств, развиваясь параллельно, способствуют перемещению приложений и информации от клиента к серверу.

Поскольку все больше пользователей получают доступ к корпоративным приложениям с помощью мобильных устройств, объем обрабатываемых данных будет расти угрожающими темпами.

Выделяют четыре основных модели реализации сред ОБ: инфраструктура как сервис (IaaS), платформа как сервис (PaaS), программное обеспечение как сервис (SaaS), бекенд как сервис (BaaS) или мобильный бекенд-сервер как услуга (mBaaS) [8].

BaaS – это облачный сервис с дополнительным функционалом, позволяющим работать с мобильными устройствами. Многие компании позволяют сотрудникам получать доступ к электронной почте и календарям, используя смартфоны и планшетные ПК. Очень немногие, однако, разрешают работать с критически-важными для компании приложениями и данными с таких устройств. И еще меньше бизнес-приложений изначально разработаны совместимыми с мобильными устройствами.

Для разработки мобильных приложений с использованием облачных сервисов необходимо найти ответы на вопросы: какие мобильные устройства будут поддерживаться; где приложение будет размещаться; создать безопасную среду для использо-

вания мобильных устройств; определить сервер BaaS, с которым будет проходить взаимодействие.

Если разрабатываемое приложение предназначено для запуска на серверах в дата-центре и там же хранить соответствующие данные, то тогда не имеет значения для каких устройств в первую очередь разработано приложение. В этом случае мобильное устройство будет иметь доступ к приложению с помощью браузера, запущенного на устройстве. Разработчики создают все более мощные приложения. Это в свою очередь требует соответствующей среды для их работы. Наиболее важным аспектом среды для функционирования мобильного приложения является архитектура этой среды. Данные мобильных приложений нужно где-то хранить, а самими приложениями нужно централизованно управлять.

Основные проблемы разработки мобильных приложений связаны с большим количеством различных типов мобильных устройств и платформ, большим количеством объемом и типов хранящихся данных, а также возрастающей угрозой безопасности. Некоторые компании разрешают сотрудникам использовать их личные мобильные устройства, чтобы не обеспечивать каждого рекомендованным устройством. Это приводит к проблемам с безопасностью, каждая из которых должна решаться централизованно. Облачный сервис BaaS становится идеальным сервисом для разработки приложений под разные мобильные устройства. BaaS также дает дополнительные возможности разработчикам, что, в свою очередь, оборачивается плюсами для пользователей. Например, пользователи обращаются к облачным приложениям через браузер, и поэтому тип мобильной ОС становится менее важным. Кроме того, благодаря высокой вычислительной мощности облака производительность мобильных приложений увеличивается.

При разработке мобильных приложений с использованием ОВ разработчик получает преимущества: быстрота разработки приложения; малое количество кода для реализации поставленных задач; экономия заряда батареи при выполнении трудоемких задач (при использовании ОВ все вычисления производятся не на клиенте а на сервере, а клиент только получает результат вычислений); экономия физической памяти телефона (весь необходимый контент для работы приложения хранится на сервере, а не на клиенте).

3. Экспериментальная проверка эффективности использования модели реализации среды облачных вычислений BaaS

Определение факторов. В исследовании в качестве фактора взяты пакеты исходных кодов ПО

для мобильных устройств на базе ОС Android, выполняющие одну задачу разными способами:

- стандартными средствами ОС Android, без использования сторонних сервисов (прототип 1);
- с использованием модулей облачного сервиса BaaS (прототип 2).

Определение откликов. Откликами являются показатели внутреннего качества ПО. Требования к внутреннему качеству определены количественно на основе метрик.

Для анализа исходного кода использована официальная среда разработки Android-приложений Android Studio [9] с установленным плагином MetricsReloaded, предоставляющим возможности расчета следующих метрик [10,11]:

- SLOC - количество строк исходного кода (Source Lines of Code, Lines of Code – LOC) является наиболее простым и распространенным способом оценки объема работ по проекту;

- DIT - глубина дерева наследования (Depth of inheritance tree), длина самого длинного пути наследования, заканчивающегося на данном модуле. Чем глубже дерево наследования модуля, тем может оказаться сложнее предсказать его поведение. С другой стороны, увеличение глубины даёт больший потенциал повторного использования данным модулем поведения, определённого для классов-предков;

- CBO - связность объектов (Coupling between objects), количество модулей, связанных с данным модулем в роли клиента или поставщика. Чрезмерная связность говорит о слабости модульной инкапсуляции и может препятствовать повторному использованию кода;

- показатель цикломатической сложности является одним из наиболее распространенных показателей оценки сложности программных проектов. Данный показатель был разработан ученым МакКейбом в 1976 г. [12], относится к группе показателей оценки сложности потока управления программой и вычисляется на основе графа управляющей логики программы (control flow graph). Данный граф строится в виде ориентированного графа, в котором вычислительные операторы или выражения представляются в виде узлов, а передача управления между узлами – в виде дуг;

- NOC - количество потомков (Number of children), число модулей, непосредственно наследующих данный модуль. Большие значения этой метрики указывают на широкие возможности повторного использования; но слишком большое значение может свидетельствовать о плохо выбранной абстракции;

- WMC - взвешенная насыщенность класса (Weighted Methods Per Class), отражает относительную меру сложности класса на основе цикломатиче-

ской сложности каждого его метода. Класс с более сложными методами и большим количеством методов считается более сложным.

Были разработаны прототипы приложений для хранения пользовательских заметок с возможностью редактирования. Прототип 1 использует реляционную базу данных для хранения данных, а в прототипе 2 данные хранятся и синхронизируются с сервером. Для работы с базой данных разработан класс DataHolder, который в прототипе 1 (рис. 1) взаимодействует напрямую с БД и передает управление программе, а во втором (рис. 2) получение и отправка данных идет через сервер.

Имя	Тип
dataHolder	DataHolder
databaseManager	DatabaseManager
noteList	List<Note>
init(Context)	void
getDataHolder()	DataHolder
updateNote(Note)	void
addNoteToNoteList(String, String, String)	void
getNoteById(int)	Note
addNewComment(int, String)	void
removeNoteFromList(Note)	void
notes	List<Note>
currentPosition	int

Рис. 1. Структура класса DataHolder прототипа 1

Имя	Тип
noteList	List<Note>
getDataHolder()	DataHolder
getNoteTitle(int)	String
getNoteDate(int)	String
getNoteComments(int)	List<String>
getNoteStatus(int)	String
getNoteId(int)	String
setNoteToNoteList(int, Note)	void
addNewComment(int, String)	void
getComments(int)	String
removeNoteFromList(int)	void
clear()	void
size()	int
addNoteToList(QBCustomObject)	void
noteListSize	int
signInUserId	int

Рис. 2. Структура класса DataHolder прототипа 2

Для статистического анализа двух выборок – воспользуемся критерием Фишера. Гипотезы:

- Н₀: между значениями параметров качества в первой выборке и второй нет существенных различий;

- Н₁: между значениями параметров качества в первой выборке и второй есть существенные различия.

Результаты анализа исходных кодов двух прототипов показаны в таблице 1.

Таблица 1

Анализ исходных кодов

№ п/п	Метрика	Прототип 1	Прототип 2
1	СВО	95	67
2	ДИТ	58	46
3	LCOM	25	24
4	RFC	4056	7846
5	NOC	3	3
6	WMC	140	105

Для подтверждения гипотезы вычислим эмпирическое значение критерия Фишера $F_{эмп}=0,00688$. Определяем по таблице критическое значение. Для уровня значимости 0,05 $F_{кр}=5,109$, а для уровня значимости 0,01 $F_{кр}=9,155$. Интерпретация результата: $F_{эмп} < F_{кр}$ для двух уровней значимости. Следовательно, гипотеза Н₀ отвергается и принимается Н₁ - между значениями параметров качества в первой выборке и второй выборке есть существенные различия.

Выводы

Результаты анализа позволяют сделать следующие общие выводы о сравнении двух разных архитектур разработки приложений для ОС Android.

1. Прототип 1 имел сложную структуру, по сравнению с прототипом 2, несмотря на использование только встроенных средств разработки приложений, но возможность работать автономно, без привязки к интернету, дает ему выигрыш над прототипом 2.

2. Возможность авторизации пользователей в прототипе 2 дает преимущества безопасности использования данных и их хранения.

3. В случае удаления приложения с устройства, прототип 1 потеряет все данные, а у прототипа 2 данные будут синхронизированы с сервисом BaaS.

4. Сопровождаемость кода прототипа 2 выше, так как его объемы меньше чем у прототипа 1. Это связано с тем, что большую часть работы берет на себя сервис BaaS, не нагружая клиента в сою очередь.

В целом же следует отметить высокое качество двух прототипов. Большинство классов имеют очень небольшой размер и сложность, а также характеризуются невысокой внешней связностью. Оба прототипы хорошо показали себя на своих функциональных возможностях.

Планируемым развитием работы является разработка методов оценки эффективности использования облачных вычислений в бизнес-приложениях.

Литература

1. Анищук, Н. Мобильность в бизнесе [Электронный ресурс] / Н. Анищук. – Режим доступа: http://www.cnews.ru/reviews/mobile_2015. – 30.08.2015.
2. Зиневич, А. Обзор рынка мобильных приложений: стоит ли игра свеч? [Электронный ресурс] / А. Зиневич. – Режим доступа: <http://megamozg.ru/post/16122>. – 4.06.2015.
3. Padilla, R. S. Components of service value in business-to-business Cloud Computing [Electronic resource] / R. S. Padilla, S. K. Milton, L. W. Johnson. – Access mode: <http://www.journalofcloudcomputing.com/content/4/1/15>. – 30.06.2015.
4. Риз, Д. Облачные вычисления [Текст] : пер. с англ. / Д. Риз. – СПб. : БХВ-Петербург, 2011. – 288 с.
5. Fenton, N. Software Metrics: A Rigorous and Practical Approach [Text] / N. Fenton, J. Bieman. – New York : Taylor & Francis Group, 2015. – 596 p.
6. Oman, P. Applying Software Metrics [Text] / P. Oman, S. L. Pfleeger. – Los Alamitos, California: IEEE Computer Society Press, 1997. – 328 p.
7. McWherter, J. Professional mobile application development [Text] / J. McWherter, S. Gowell. – Indianapolis: John Wiley & Sons Inc., 2012. – 432 p.
8. Kavis, M. J. Architecting the Cloud: Design Decision for Cloud Computing Service Models (SaaS, PaaS, & IaaS) [Text] / M. J. Kavis. – Hoboken, New Jersey: John Wiley & Sons Inc., 2014. – 200 p.
9. Android SDK [Electronic resource] <http://developer.android.com/index.html>. – 30.08.2015.
10. Kaner, C. Software Engineering Metrics: What Do They Measure and How Do We Know? [Text] / C. Kaner, W. P. Bond. // 10TH International Software Metrics Symposium, Metrics 2004. – P. 1-12.
11. Laird, L. Software Measurement and Estimation: a practical approach [Text] / L. Laird, M. Carol Brennan. – New Jersey : John Wiley & Sons, Inc., Hoboken, 2006. – 257 p.
12. McCabe, T. J. A Complexity Measure [Text] / T. J. McCABE. // IEEE Transactions On Software Engineering. – December 1976. – Vol. SE-2, NO. 4. – P. 308-320.

Поступила в редакцию 30.10.2015, рассмотрена на редколлегии 18.11.2015

ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ ХМАРНИХ ОБЧИСЛЕНЬ У БІЗНЕС-ДОДАТКАХ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID

Є. В. Соколова

Розглянуто використання моделей середовищ реалізації хмарних обчислень у бізнес-додатках для мобільних пристроїв. Обґрунтовано вибір хмарної серверної інфраструктури BaaS для мобільних додатків. Визначено особливості розробки програмного забезпечення для мобільних пристроїв з використанням хмарних обчислень. Описано експеримент для перевірки ефективності використання моделі реалізації середовища хмарних обчислень BaaS. Проведено аналіз експериментально отриманих результатів, виявлено негативні й позитивні сторони використання хмарних обчислень у розробці бізнес-додатків для мобільних пристроїв на базі ОС Android.

Ключові слова: програмне забезпечення, хмарні обчислення, мобільні пристрої, метрики вихідного коду.

EXPERIMENTAL EVALUATION OF CLOUD COMPUTING IN THE BUSINESS APPLICATIONS BASED ON OPERATING SYSTEM ANDROID

E. V. Sokolova

The use of models of implementation environments of cloud computing in business applications for mobile devices has been considered. The choice of cloud server infrastructure BaaS for mobile applications has been grounded. The features of software development for mobile devices with using cloud computing has been identified. The experiment for the verification of the effectiveness of using the model of implementation environment of cloud computing BaaS has been described. The analysis of the experimental results has been conducted, the negative and positive aspects of using the cloud computing in the development of business applications for mobile devices based on Android OS has been detected.

Key words: software, cloud computing, mobile devices, source code metrics.

Соколова Евгения Витальевна – канд. техн. наук, доцент каф. инженерии программного обеспечения, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: EVS_khai@gmail.com.