

УДК 004.519.217

Д. А. МАЕВСКИЙ, Е. Д. МАЕВСКАЯ, А. В. ДРОЗД

*Одесский национальный политехнический университет, Украина***МЕТРИКИ «ОТТЕНКОВ ЗЕЛЕНОГО» В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ**

В работе предложен набор метрик, с помощью которых можно выполнить косвенную оценку программного обеспечения по показателям эффективности использования оперативной памяти и процессорного времени. Показано, что эти показатели характеризуют общие энергетические затраты при выполнении программ. Оценка эффективности позволяет выбрать наиболее оптимальное программное обеспечение, удовлетворяющее парадигме «зеленых» технологий. Предложенные метрики могут использоваться как для традиционного программного обеспечения, так и для выбора наиболее оптимального описания архитектуры ПЛИС на языке VHDL.

Ключевые слова: зеленое программное обеспечение, зеленые технологии, метрики эффективности, минимизация ресурсов, качество программного обеспечения, стандарт качества, модель качества.

Введение

Сегодня человечество пришло к пониманию того естественного факта, что ресурсы, которые оно потребляет для обеспечения своей деятельности, не бесконечны. В первую очередь это относится к энергетическим ресурсам, таким как газ, нефть и другие виды топлива. Кроме того, производство энергии зачастую связано с негативным влиянием на экологию планеты, причем это влияние все время возрастает.

Одним из путей решения проблемы энергетических ресурсов может заключаться в развитии технологий, позволяющих экономить энергию – так называемых «зеленых» технологий. Бурное развитие этих технологий мы наблюдаем сегодня.

Средства вычислительной техники, компьютерные системы, сами по себе не относятся к энергоемким. Ну что такое, казалось бы, 500 ватт мощности, потребляемой одним персональным компьютером? Стоит ли о них беспокоиться? Оказывается, стоит, если учесть общее количество компьютеров в мире. А эта цифра сегодня приближается к нескольким миллиардам. Поэтому закономерно, что зеленые технологии проникли и в IT-отрасль.

Программное обеспечение, под управлением которого работает любая компьютерная система, также может способствовать сокращению энергопотребления. Для того, чтобы стать зеленым, программное обеспечение должно удовлетворять следующим требованиям:

1. Требование минимизации ресурсов, необходимых для функционирования программного обеспечения. К таким ресурсам относится объем оперативной памяти компьютера, тактовая частота и

быстродействие центрального процессора.

2. Требование экономии ресурсов, потребляемых периферийными устройствами, которыми управляет программное обеспечение. К таким ресурсам напрямую относится объем потребляемой энергии и связанные с ним выбросы вредных веществ в атмосферу.

3. Требование к надежности функционирования и устойчивости по отношению к внешним целевым воздействиям, направленным либо на прекращение функционирования, либо на получение контроля над программным обеспечением.

Однако, если мы хотим управлять каким-либо процессом, то сначала нужно определить измеряемые характеристики этого процесса, контролируя которые мы сможем судить об эффективности управляющих воздействий. Поэтому, если мы хотим создавать зеленое программное обеспечение – следует ввести однозначно определяемые метрики, на основании которых можно будет судить о том, что созданная программа действительно зеленая. То есть, фактически, такие метрики должны показывать оттенки степени «зелености» программного обеспечения.

Рассмотрим основные характеристики качества программного обеспечения и их соответствие перечисленным требованиям.

1. Современное состояние проблемы

Понятие качества программного обеспечения и основные показатели качества закреплены в стандарте ISO/IEC 25010-2011 [1]. Этим стандартом вводится важное для дальнейшего рассмотрения понятие «модель качества программных систем»,

которая включает определенные наборы характеристик и подчиненных им так называемых «субхарактеристик». Каждая из субхарактеристик должна иметь числовое значение, которое устанавливается благодаря введению специальной метрики для этой субхарактеристики. Основные характеристики, определяющие качество программного обеспечения, перечислены ниже.

– Функциональность (Functionality) – включает набор атрибутов, показывающих существование определенных функций ПО с их заданными свойствами. Атрибуты показывают, насколько эти функции удовлетворяют заявленным или предсказуемым потребностям;

– Надежность (Reliability) – набор атрибутов, которые показывают способность программного обеспечения сохранять свой уровень производительности при заданных условиях в течение заданного периода времени;

– Безопасность (Security) – набор атрибутов, характеризующих степень конфиденциальности и сохранности информации, а также отказоустойчивость и возможность подтверждения аутентичности ПО;

– Полезность (Usability) – набор атрибутов, которые указывают как на усилия, необходимые для использования ПО, так и на индивидуальные оценки такого использования возможными пользователями;

– Эффективность работы (Efficiency) – набор атрибутов, которые имеют отношение к взаимосвязи между уровнем производительности ПО и количеством ресурсов, используемых при заданных условиях;

– Способность к сопровождению (Ремонтпригодность – Maintainability) – набор атрибутов, показывающих усилия и ресурсы, которые необходимы, чтобы сделать требуемые изменения на этапе эксплуатации;

– Мобильность (Portability) – набор атрибутов, которые показывают способность программного обеспечения быть перенесенным из одной операционной среды в другую.

Как видим, требования к зеленому программному обеспечению стандартом ISO/IEC 25010-2011 четко не выделяются.

Требование 1, относительно минимизации ресурсов, необходимых для функционирования программного обеспечения, частично покрывается характеристикой «Эффективность использования», которая согласно стандарту, имеет такие субхарактеристики, как «Использование ресурсов» и «Мощность». Последняя субхарактеристика относится, очевидно, не к самому программному обеспечению, а к аппаратуре, на которой оно функционирует и

перекрывается субхарактеристикой «Использование ресурсов».

Требование 3, в плане надежности функционирования покрывается характеристикой «Надежность», а в плане устойчивости частично покрывается характеристикой «Безопасность», которая имеет субхарактеристики «Целостность» и «Безотказность». Частичность покрытия обусловлена тем, что устойчивость к негативным целевым воздействиям только лишь связана с целостностью ПО, но не обусловлена ею. Безотказность же – это скорее следствие, а не причина устойчивости.

Требование 2, относительно экономии ресурсов, потребляемых работающими под управлением программного обеспечения периферийными устройствами, в стандарте ISO/IEC 25010:2011 не отражено вообще.

Таким образом, можно видеть, что в стандарте ISO/IEC 25010:2011 нет четко определенных характеристик, позволяющих определять степень «зелености» программного обеспечения. В связи с этим, дальнейшими задачами исследований в области стандартизации зеленого программного обеспечения должны стать:

1. Разработка метрик «оттенков зеленого», которые в числовом виде отражали бы способность программного обеспечения к экономии и сохранению используемых ресурсов.

2. Разработка новой модели качества программного обеспечения с учетом метрик «оттенков зеленого». В этой новой модели качества должны быть определены не только состав характеристик и субхарактеристик качества зеленого ПО, но и даны рекомендации по получению их численных значений.

Настоящая работа посвящена решению первой задачи – разработке системы метрик, отражающих способность программы к экономии и сохранению ресурсов.

2. Метрики «оттенков зеленого»

Основными ресурсами, потребляемыми программой, являются объем оперативной памяти и время работы центрального процессора. Эти показатели косвенно характеризуют энергетические затраты на функционирование программы. Понятно, что чем дольше работает программа, тем больше энергии она потребляет. Поэтому метрические характеристики должны учитывать эффективность применяемых алгоритмов и их программной реализации.

Связь объема оперативной памяти с энергопотреблением на первый взгляд не так очевидна. Для того, чтобы разобраться с этим вопросом, вспомним,

что оперативная память представляет собой набор элементарных ячеек, каждая из которых хранит один бит информации. Ячейка состоит из одного полевого транзистора и одного конденсатора [2]. Для изменения состояния ячейки памяти необходимо изменить заряд этого конденсатора. Изменение заряда происходит не только при изменении состояния, но и при простом считывании состояния. Как известно, при считывании заряд полностью теряется, а для его восстановления выполняется процедура регенерации памяти.

Заряд конденсатора мгновенно изменить нельзя – об этом говорит известный закон коммутации в теоретической электротехнике. То есть, изменение заряда происходит не мгновенно, а в течение какого-то промежутка времени. Но ведь скорость изменения заряда – это и есть электрический ток. А для его поддержания необходимо затратить определенную энергию. Таким образом, неэкономно расходуя оперативную память и не оптимально организуя обращение к ней, программное обеспечение косвенно способствует увеличению потребления электрической энергии.

Итак, метрики оттенков зеленого должны учитывать затраты памяти и процессорного времени. с учетом этого, разработанные метрики можно условно разделить на три группы:

- метрики эффективности использования памяти;

- метрики эффективности использования процессорного времени;
- смешанные метрики, оценивающие память и время одновременно.

Предложенные метрики эффективности использования памяти приведены в таблице 1.

Пояснения требует показатель M01 - количество неиспользуемых в программе методов классов. Современные системы программирования, особенно объектно-ориентированные, практически полностью скрывают от программиста механизмы управления оперативной памятью, а стандартные библиотеки включают в один класс десятки, если не сотни методов., что требует значительного объема памяти.

Показатель M05 определяет количество заново объявленных переменных в программе. Для того чтобы избежать неэффективного использования памяти следует один раз объявить переменную в начале программы и использовать ее постоянно.

Показатель M06 отражает ситуацию, при которой переменная, по логике алгоритма, прекращает использоваться до окончания программы, но при этом вводится новая переменная, с другим именем, но того же самого типа.

Метрики эффективности использования процессорного времени представлены в таблице 2.

Здесь пояснений требует показатель T07, определяющий количество вложенных циклов с невоз-

Таблица 1

Метрики эффективности использования оперативной памяти

Индекс	Показатель	Пояснение
M01	Количество не используемых в программе методов классов	Оценивается резервируемая, но не используемая программой оперативная память
M02	Размер неиспользуемой памяти в статически определенных массивах	
M03	Размер никогда не выполняющихся участков программного кода	Занимаемая, но неиспользуемая память
M04	Количество повторно используемых одинаковых по значению текстовых или нецелочисленных констант	Память, используемая неэффективно
M05	Количество переменных, заново объявленных в тексте программы	
M06	Количество переменных, использование которых прекращено до завершения текста программы (отдельно по типам переменных)	Память, высвобождаемая с запаздыванием

Таблица 2

Метрики эффективности использования процессорного времени

Индекс	Показатель	Пояснение
T01	Количество операторов умножения и деления целочисленных переменных на степень двойки	Неоправданное использование ресурсоемких средств
T02	Количество операторов возведения в целую степень при помощи встроенных функций	
T03	Суммарное количество измерений во всех статических массивах	Средства, потенциально увеличивающие время выполнения
T04	Общее количество операторов цикла в программе	
T05	Суммарный уровень вложенности всех циклов программы	

Окончание табл. 2

Индекс	Показатель	Пояснение
T06	Количество операторов в теле цикла, не зависящих от значения переменной цикла	Ненужные повторные вычисления
T07	Количество вложенных циклов с невозрастающим по уровням вложенности числом повторений (если можно определить при статическом анализе программы)	Средства, увеличивающие время выполнения
T08	Количество явных или неявных преобразований типов переменных в выражениях вне операторов цикла	Операции, требующие большого времени выполнения
T09	Количество явных или неявных преобразований типов переменных в выражениях в теле операторов цикла	
T10	Общее количество операций с плавающей точкой	
T11	Общее количество операций деления, хотя бы один из операндов которых имеет тип с плавающей точкой	
T12	Количество переменных, иницируемых при помощи оператора присваивания	
T13	Количество последовательно расположенных операторов ввода-вывода на одно и то же устройство	Неэффективное использование ввода-вывода
T14	Количество последовательно расположенных коротких условных операторов (без ELSE) со сравнением одной и той же переменной	Использование оператора CASE более эффективно

растающим по уровням вложенности числом повторений, что приводит к неэффективному использованию времени выполнения.

Группа показателей T08 – T12 определяет операции, требующие большого времени выполнения. Сюда относятся, во-первых, операции явного или неявного преобразования типов. Особенно это относится к преобразованиям типов, выполняемым внутри операторов циклов. Во-вторых, по возможности следует избегать операций с плавающей точкой. Даже при наличии встроенных математических сопроцессоров, эти операции являются очень затратными по времени.

Кроме перечисленных в таблицах 1 и 2, можно выделить еще несколько смешанных метрик, оценивающих одновременно как затраты памяти, так и процессорного времени. Эти метрики приведены в таблице 3.

Показатель TM01 определяет редкий, но возможный случай, когда в параметрах процедур или функций массивы передаются не по ссылке, а по значению. Согласно правилам передачи по значе-

нию, в процедуру передается копия массива, создание которой требует затрат как времени, так и оперативной памяти.

Тех же затрат требует и выполнение рекурсивных процедур. Не смотря на то, что современные языки программирования практически всегда допускают создание рекурсивных процедур, лучше от использования рекурсии вообще отказаться.

Выводы

В настоящей статье предложены метрики, позволяющие оценить степень «зелености» программного обеспечения. Использование этих метрик позволяет косвенно оценить затраты времени и оперативной памяти при работе программного обеспечения. Данные метрики могут применяться для сравнительной оценки однотипного по функционалу программного обеспечения с целью выбора наиболее оптимальной реализации.

Таблица 3

Метрики эффективности использования времени выполнения и памяти

Индекс	Показатель	Пояснение
TM01	Количество массивов в параметрах процедур, передаваемых по значению	Происходит копирование массива
TM02	Количество рекурсивных процедур	Рекурсия требует дополнительных затрат и времени и памяти
TM03	Суммарный уровень рекурсии (если можно определить при статическом анализе текста)	

Следует отметить, что указанный подход может использоваться не только для программ, написанных на обычных языках программирования.

С незначительными переработками его можно использовать и для оценки текстов на языке VHDL, который используется для описания архитектуры интегральных схем [3]. Выбор с использованием метрик наиболее оптимального описания позволит создавать оптимальные по энергетическим затратам микросхемы, особенно при программировании ПЛИС [4].

Следует отметить, что предложенный набор метрик не является исчерпывающим и может меняться в зависимости от особенностей конкретного языка программирования и конкретной вычислительной системы.

Оценка, выполненная при помощи указанных метрик, является косвенной, то есть с ее помощью невозможно определить абсолютные, выраженные в ваттах, значения расходуемой энергии.

Поступила в редакцию 18.02.2014, рассмотрена на редколлегии 25.03.2014

Рецензент: д-р техн. наук, проф. И. Б. Туркин, Национальный аэрокосмический университет «ХАИ», Харьков, Украина.

Литература

1. ISO/IEC 25010:2011. *Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. [Electronic resource]. Access mode: <http://www.standards.ru/document/4580604.aspx>
2. Скотт, М. Модернизация и ремонт ПК = *Upgrading and Repairing PCs* [Текст] / Мюллер Скотт. – М.: Вильямс, 2007. – 1072 с.
3. Бибило, П. Н. Основы языка VHDL [Текст]: учебное пособие / П. Н. Бибило. – М.: Книжный дом «ЛИБРОКОМ», 2012. – 328 с.
4. Грушевицкий, Р. И. Проектирование систем на микросхемах с программируемой структурой [Текст] / Р. И. Грушевицкий, А. Х. Мурсаев, Е. П. Угрюмов. – СПб.: БХВ-Петербург, 2006. – 736 с.

МЕТРИКИ «ВІДТІНКІВ ЗЕЛЕНОВОГО» В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ

Д. А. Маєвський, О. Д. Маєвська, О. В. Дрозд

У роботі запропонований набір метрик, за допомогою яких можна виконати непряму оцінку програмного забезпечення за показниками ефективності використання оперативної пам'яті та процесорного часу. Показано, що ці показники характеризують загальні енергетичні витрати при виконанні програм. Оцінка ефективності дозволяє вибрати найбільш оптимальне програмне забезпечення, що задовольняє парадигмі «зелених» технологій. Запропоновані метрики можуть використовуватися як для традиційного програмного забезпечення, так і для вибору найбільш оптимального опису архітектури ПЛИС на мові VHDL.

Ключові слова: зелене програмне забезпечення, зелені технології, метрики ефективності, мінімізація ресурсів, якість програмного забезпечення, стандарт якості, модель якості.

METRICS OF "SHADES OF GREEN" IN THE SOFTWARE

D. A. Maevsky, E. D. Maevskaya, A. V. Drozd

The paper proposes a set of metrics that can be used to perform an indirect assessment of the software in terms of efficient use of memory and the Processing time. It is shown that these parameters characterize the overall energy costs for running programs. Allows assessment of the effectiveness of programs to choose the optimal bonus software that satisfies the paradigm of "green" technologies. The proposed metrics can be used for both traditional software and to select the most optimal FPGA architecture description on VHDL language.

Keywords: green software, green technology, performance metrics, minimizing resources, software quality, quality standards, quality model.

Маєвський Дмитрій Андреевич – д-р техн. наук, доцент, заведуючий кафедрой теоретических основ и общей электротехники Одесского национального политехнического университета, г. Одесса, Украина, e-mail: Dmitry.A.Maevsky@gmail.com.

Маєвська Елена Дмитриевна – студент Одесского национального политехнического университета, г. Одесса, Украина, e-mail: lmd29@ukr.net.

Дрозд Александр Валентинович – д-р техн. наук, профессор, профессор кафедры компьютерных интеллектуальных систем и сетей Одесского национального политехнического университета, г. Одесса, Украина, e-mail: drozd@ukr.net.