

О МЕТОДЕ ОЦЕНКИ СООТВЕТСТВИЯ ЭНЕРГОЗАТРАТНОСТИ ВЫЧИСЛИТЕЛЬНОЙ ЗАДАЧИ ТРЕБОВАНИЯМ К ЭНЕРГОПОТРЕБЛЕНИЮ И ЕГО РЕАЛИЗАЦИИ С ПОМОЩЬЮ StarPU

К.М. Ханкин

METHOD OF CONFORMANCE EVALUATION OF ENERGY CONSUMPTION OF COMPUTING TASK AND REQUIREMENTS TO ENERGY CONSUMPTION, ITS IMPLEMENTATION WITH THE HELP OF StarPU

K.M. Khankin

Приведено описание возможного метода для проверки соответствия энергозатратности вычислительной задачи требованиям к энергопотреблению, выставляемым целевой вычислительной системой, а также описан подход к реализации приведённого метода с использованием StarPU.

Ключевые слова: StarPU, энергопотребление.

In the article the description of a possible method for checking a compliance of computing task energy consumption level to the consumption limits of computing system and its implementation with the help of StarPU are considered.

Keywords: StarPU, energy consumption.

Введение

В современных объектах информатизации и народного хозяйства актуальной является проблема снижения потребления энергии. Ведущие производители бытовой, компьютерной и прочей техники разрабатывают различные технологии энергосбережения, например [1]. Разрабатываются мини-платформы, обладающие, с одной стороны, низким энергопотреблением, а с другой стороны – вычислительной мощностью, сравнимой с настольными системами [2]. Однако не менее актуальной является проблема проверки соответствия задач установленному ограничению на энергозатраты. Например, эта задача актуальна для спутниковых систем, где ресурсы ограничены энергией, вырабатываемой системой электроснабжения на солнечных батареях, например [3]. Для систем с ограничениями на максимум энергопотребления важно убедиться в том, что задача, возложенная на сис-

тему, может быть этой системой выполнена до того, как запустить систему в эксплуатацию.

В источниках описывается метод оценки энергопотребления задач путём её пробного исполнения. Например, Intel предлагает следить за разрядом батареи ноутбука [4]. Однако не всегда возможно оценить энергозатраты таким способом, поэтому появляется необходимость в инструментах оценки энергозатратности задания. Для создания таких инструментов необходимо предварительно выработать методы проведения «энергетической» оценки кода задачи.

Один из подходов к оценке энергозатрат

В исходном коде задачи можно выделить основные алгоритмические единицы – циклы, условные операторы и т. д. Можно предположить, что энергозатраты на исполнение программы будут примерно равны (или меньше за счёт оптимиза-

Ханкин Константин Михайлович – аспирант кафедры ЭВМ, Южно-Уральский государственный университет; hc@comp.susu.ac.ru

Khankin Konstantin Mikhaylovich – Post-Graduate Student of Electronic Computer Department of South Ural State University; hc@comp.susu.ac.ru

ций) сумме энергозатрат на выполнение отдельных алгоритмических единиц. Таким образом, один из возможных подходов заключается в энергетической оценке каждой алгоритмической единицы и суммировании этих оценок. Рассмотрим, как можно оценить энергозатраты каждой алгоритмической единицы.

Вообще для оценки энергозатрат требуется предварительно знать, какую мощность потребляет вычислительная система на единичную операцию. Это можно выяснить, выполняя элементарные операции (сложение, умножение, деление и пр.) и снимая данные об энергопотреблении во время их выполнения. Причём если вычислительная система имеет несколько вычислителей, а возможно, и разнородных, то такую оценку производить надо для каждого вычислителя на таких наборах данных, которые являются типичными для данного вычислителя (например, для оценки CUDA-совместимого вычислителя следует учитывать размеры вычислительной сетки и количество процессоров).

Без оптимизаций последовательный код можно представить в виде набора элементарных команд и соответственно оценить энергозатраты на его выполнение как сумму энергозатрат на выполнение каждой элементарной единицы. При оценке условного оператора целесообразно оценивать все ветки, разворачивая условие в последовательный код. Полученные оценки укладываются в некоторый диапазон, который и будет являться результатом оценки. Однако при оценке циклов всё не так очевидно. Если цикл имеет ограниченное число итераций, то оценить его энергоёмкость легко по методу, применимому для условных операторов, развернув такой цикл в последовательный код. Если вся программа заключена в бесконечный цикл (как, например, основной цикл в микропроцессорных системах), можно оценить энергоёмкость одной итерации такого цикла и, зная время, за которое исполняется эта итерация, получить оценку. Для циклов с заранее неизвестным или переменным числом итераций можно разработать эвристики, которые будут давать оценку с более высокой погрешностью, чем для других алгоритмических единиц, но всё же позволят оценить всю задачу целиком.

Ещё более сложным является анализ исходного кода, включающего в себя библиотеки, исходного кода которых мы не имеем. В случае с динамическим связыванием нельзя будет даже оценить энергозатраты по бинарному коду. Для такого варианта предлагается сделать набор готовых энергооценок часто применяемых функций из различных библиотек путём предварительного выполнения тестовых прогонов на разных объёмах данных с целью выявления зависимости энергопотребления от количества и размера аргументов. В случае, если для какой-то функции не окажется оценки в тот момент, когда она будет необходима, надо бу-

дет выполнить такой тест до исполнения поступившей задачи.

Применение StarPU для оценки

Для реализации описанного выше подхода можно применять систему диспетчеризации блоков вычислительной задачи StarPU. StarPU при первом запуске выполняет оценку быстродействия всех вычислителей в системе для реализации алгоритма диспетчеризации. Можно совместить эту оценку с оценкой энергозатрат на элементарные операции, если оснастить вычислительную систему оборудованием для измерения энергопотребления, аналогичным использованному в [5], и добавить в StarPU возможность использования измененной величины. Дальнейшую работу StarPU выполняет на основе одного из алгоритмов диспетчеризации, критерии для которого записаны в блоках кода. Можно добавить дополнительный критерий – энергозатратность, который будет либо вычисляться (оцениваться), либо задаваться пользователем и модифицировать алгоритм диспетчеризации, изменив целевую функцию на достижение минимума энергопотребления. При этом можно оставить вторым критерием оптимизации минимум времени выполнения задачи, получив тем самым решение, оптимальное по энергопотреблению в первую очередь, и по времени выполнения – во вторую. StarPU включает в себя множество алгоритмов диспетчеризации [6] и позволяет реализовывать собственные диспетчеры, что даёт возможность выбирать целевую функцию на этапе выполнения задачи.

Выводы

Для проверки возможности выполнения задачи с затратой предложенного объёма энергии можно использовать энергооценку исходного кода задачи. Для такой оценки код можно разбить на элементарные операции и алгоритмические конструкции и, используя имеющуюся энергетическую оценку операций и конструкций, оценить энергозатратность задания, сделав вывод о возможности его выполнения на предложенном объёме энергоресурса. Для автоматизации этого процесса можно использовать StarPU, изменив алгоритмы диспетчеризации блоков кода и добавив к блокам новый критерий диспетчеризации – «энергоёмкость».

Литература

1. *Low Power Embedded Systems with Intel Power Management Technology / Intel Corporation* – http://www.intel.com/p/ru_RU/embedded/hws/technology/power-management
2. *Raspberry Pi. An ARM GNU/Linux box.* – <http://www.raspberrypi.org/>
3. *Системы электроснабжения / КБ «Южное».* – http://www.yuzhnoye.com/?id=58&path=Aerospace%20Technology/aerospace_systems_and_

*components/sc_components/power_supply_systems/
power_supply_systems*

4. *Intel Developer Zone. Способы оценки энергопотребления программы / Intel Corporation – http://software.intel.com/ru-ru/articles/Energy_Consumption_Tools*

5. Ханкин, К.М. Оценка влияния аппаратных технологий энергосбережения персонального ком-

пьютера на производительность и энергопотребление / К.М. Ханкин // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2012. – Вып. 16. – № 23 (282). – С. 225–227.

6. *Task scheduling policy. StarPU Handbook. – <http://runtime.bordeaux.inria.fr/StarPU/starpu.html#Task-scheduling-policy>*

Поступила в редакцию 4 декабря 2012 г.