

УДК 004.414.22/004.052

И. Б. ТУРКИН, А. В. ВДОВИТЧЕНКО

*Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина***МОДЕЛЬ АКТИВНОСТИ ПОЛЬЗОВАТЕЛЯ В ПОЛИТИКЕ УПРАВЛЕНИЯ ЭНЕРГОПИТАНИЕМ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА**

В статье проанализированы существующие методы управления питанием персонального компьютера. Рассмотрена модель управления энергопитанием персонального компьютера и ее реализации в современных операционных системах. Предложена модель активности пользователя, основанная на скрытой Марковской модели. Активность пользователя рассматривается как на синтаксическом, так и на семантическом уровне. Разработана структура модели активности пользователя. Предложена и обоснована параметрическая идентификация модели методом Баума-Уэлча. Модель активности пользователя позволит адаптировать политику динамической системы управления энергопотребления с учетом фактора пользователя.

Ключевые слова: энергопотребление, управление питанием, пользователь, политика энергопотребления, ACPI, скрытые Марковские модели, метод Баума-Уэлча.

Введение

Потребляемая мощность персональных компьютеров стала серьезной проблемой в последние годы. Энергетический коэффициент полезного действия, необходимость продлить срок службы батареи, а также снизить эксплуатационные расходы и воздействие на окружающую среду являются актуальными вопросами во многих научных сообществах. Существует несколько подходов, которые позволяют экономить энергию за счет адаптации к изменениям в окружающей среде: динамическое управление питанием (DMP – Dynamic power management) [1] и динамическое масштабирование напряжения (DVS – Dynamic voltage scaling) [2].

Модель управления энергопотреблением основывается на четырех основных сущностях: пользователь, прикладные программы, операционная система и аппаратная часть. Последние две сущности и частично вторая получили свою реализацию в виде схем энергопотребления в операционных системах, в частности и в ОС Windows, которые основаны на DVS [3] с одной стороны, где выделяют три следующие схемы:

– высокой производительности, когда отключено динамическое масштабирование производительности в соответствии с рабочей нагрузкой, а вместо этого предоставлены уровни постоянного повышения производительности при соответствующем росте энергопотребления;

– сбалансированная схема, когда реализуется динамическое масштабирование производительности в зависимости от требований текущей рабочей

нагрузки;

– энергосберегающая схема предназначена для максимальной экономии энергии и хорошо подходит для увеличения времени автономной работы в мобильных ПК.

С другой стороны управление энергопотреблением основывается на DMP, а именно переходе между ACPI-состояниями системы. Каждое состояние системы характеризуется уровнем производительности и потребления энергии. Глобальные состояния системы представлены в таблице 1.

Таблица 1

Состояния системы ACPI

Название состояния	Описание состояния
G0 (S0) (Рабочее)	Выполнение стандартных операций. Максимальная производительность при обычном энергопотреблении
G1 (приостановлено, спящее, гибридный сон)	Пониженное энергопотребление с возможностью быстрого восстановления рабочего состояния, без потери данных
G2 (S5) (программное выключение)	Выключение операционной системы при незначительном потреблении энергоресурсов, возможность более быстрого запуска по сравнению с «холодным стартом» состояния G3
G3 (Выключенное)	Полное отключение, не потребляет энергии

Большинство текущих политик основывается на измерении времени бездействия, когда пользователь сам задает время, по истечению которого, при условии бездействия, система перейдет в одно из энергосберегающих состояний (см. табл. 1) [4,5]. Но такая политика является статической, что приводит к медленному реагированию на изменения состояния пользователя и не позволяет адаптироваться к текущей ситуации.

Таким образом, **целью статьи** является разработка модели активности пользователя в системе динамического управления питанием, которая позволяет динамично реагировать на изменения состояния пользователя, для снижения энергопотребления ПК.

1. Модель активности пользователя

Предложим иную политику, основой которой будет модель поведения пользователя, определяемая по текущим наблюдаемым и измеряемым величинам состояния системы в момент времени t . Модель поведения пользователя (MU) предполагает представление истории активности пользователя на 2 уровнях: синтаксическом и семантическом:

$$MU = \langle S_y, S_e \rangle, \quad (1)$$

при этом синтаксический уровень системных событий – это уровень взаимодействия пользователя с клавиатурой, мышью, буфером обмена, а семантический уровень характеризует управление пользователем запуском/остановом приложений, переключением фокуса и т.д.

Метриками активности пользователя и на синтаксическом, и на семантическом уровнях являются количества событий в единицу времени.

Для семантического уровня, чтобы получить взвешенную оценку, классифицируем работу с приложениями на группы.

Группа 1. Программные средства общего назначения (текстовые редакторы, текстовые процессоры, системы компьютерной верстки, графические редакторы, электронные таблицы, веб-браузеры) – взвешивающий коэффициент равен 1.

Группа 2. Программные средства развлекательного назначения (медиа плееры, компьютерные игры) – взвешивающий коэффициент равен 1.5.

Группа 3. Профессиональные программные средства – взвешивающий коэффициент равен 1.2.

В результате проведения эксперимента по оценки активности пользователя во времени с учетом используемых приложений было выделено три основных качественных состояния поведения пользователя – это низкая активность, средняя активность и высокая активность (рис. 1).

Будем полагать, что прогнозируемое, будущее состояние пользователя зависит только от того состояния, в котором пользователь находится сейчас. Мы не можем наблюдать состояния пользователя, так как они скрыты, наблюдаемыми являются только отклики. Исходя из выше сказанного, для описания модели поведения пользователя будем использовать аппарат скрытых Марковских сетей. Обобщенная структура скрытой Марковской модели представлена на рисунке 2.

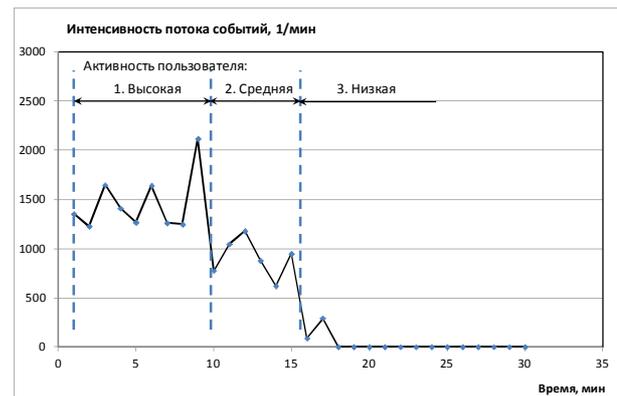


Рис. 1. Экспериментальная оценка активности пользователя

В общем случае модель поведения пользователя (MAU) можно представить как:

$$MAU = \langle S, O, A, B \rangle, \quad (2)$$

где S – это набор скрытых состояний $S = \{1, \dots, N\}$;

O – количество наблюдаемых переменных-откликов;

$A = \{a_{ij}\} = P(S_t = j | S_{t-1} = i)$ – матрица условных вероятностей перехода между состояниями S в момент времени t ;

B – матрица эмиссионных вероятностей.

Видимый слой скрытой модели Маркова (СММ) представлен наблюдаемыми переменными. То есть активностью пользователя, которая измеряется интенсивностью событий за определенный интервал времени с учетом группы используемых приложений. Так же в видимый слой вынесены два наблюдаемых состояния скрытого слоя – это начало и завершение работы, когда активность пользователя равна нулю.

Скрытый слой СММ отображает состояния, которые измеряются интервалом времени до завершения работы с персональным компьютером. Выделим следующие скрытые состояния: 5-10 минут; 10-15 минут; 15-20 минут; 20-25 минут. Будем считать, что интервалы меньше 5 мин и больше 25 являются не диагностируемыми.

Необходимость пар. Идентификации СММ.

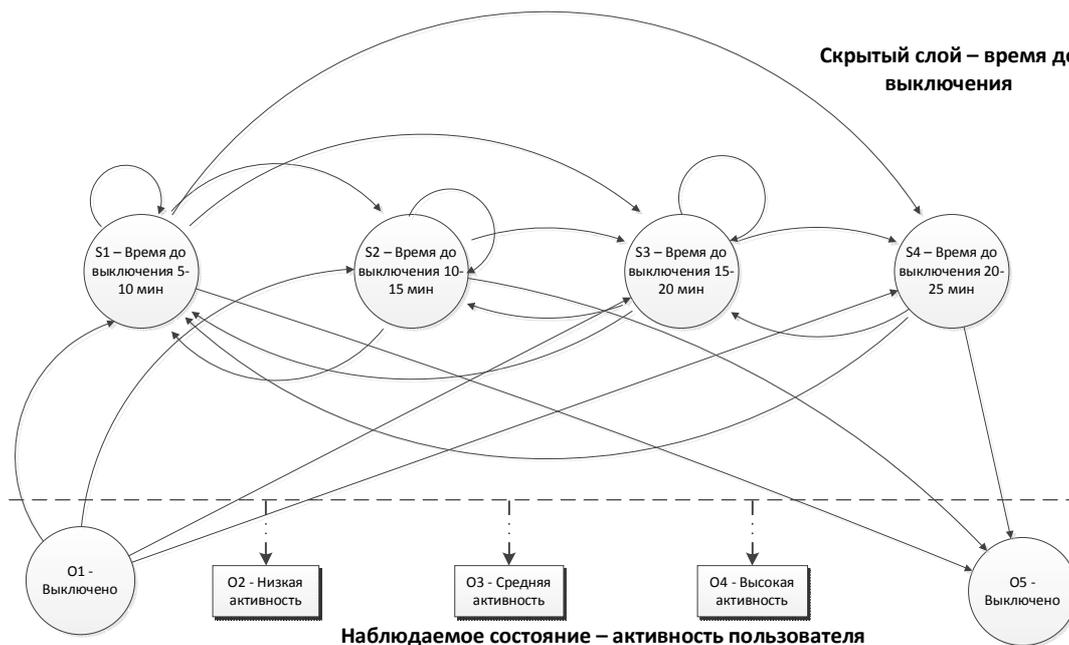


Рис. 2. Структура скрытой Марковской модели

2. Параметрическая идентификация модели методом Баума-Уэлча

Существует два метода параметрической идентификации модели:

- метод Витерби использует оптимальный путь, является быстрым. Для работы требуется знать точное распределение наблюдаемых характеристик и модель [6].

- метод Баума-Уэлча (максимизации ожидания) [7], итеративный метод, позволяющий найти локальный максимум функции вероятности $p(O|MAU)$.

Для параметрической идентификации использован метод Баума-Уэлча, так как данный метод гарантирует, что модель с новыми параметрами будет равна или лучше предыдущей модели с точки зрения критерия максимального правдоподобия. Суть данного метода заключается в том, что сначала считаем ожидание наблюдений, при условии заданной модели, затем корректируем модель в зависимости от имеющихся наблюдений видимого слоя. Метод использует алгоритм прямого-обратного хода и является частным случаем обобщенного expectation-maximization алгоритма [8].

Данный метод предполагает наличие следующих исходных данных:

- структуры модели;
- матрицы условных вероятностей A ;
- матрицы эмиссионных вероятностей B ;

- вектора начального распределение вероятностей нахождения в состояниях π .

Существует несколько подходов к выбору исходных данных, а именно на основе априорных знаний о предметной области протекания процесса или случайным образом. Так как результатом метода является локальный максимум правдоподобия модели, то от выбора начального распределения зависит конечное правдоподобие.

Исходные данные метода Баума-Уэлча:

$$MAU = (A, B, \pi). \quad (3)$$

Алгоритм метода Баума-Уэлча итеративно обновляет параметр MAU до схождения в одной точке.

Прямой ход алгоритма:

$\alpha_i(t) = p(O_1 = o_1, \dots, O_t = o_t, S_t = i | MAU)$, что является вероятностью получения заданной последовательности $\{o_1, \dots, o_t\}$ для состояния i в момент времени t .

$\alpha_i(t)$ вычисляем рекурсивно:

$$\alpha_i(1) = \pi_i \cdot b_i(O_1);$$

$$\alpha_j(t+1) = b_j(O_{t+1}) \sum_{i=1}^N \alpha_i(t) \cdot a_{ij}.$$

Обратный ход алгоритма:

Данная процедура позволяет вычислить вероятность конечной последовательности o_{t+1}, \dots, o_T при условии, что начинаем из исходного состояния i в момент времени t .

$\beta_i(t)$ вычисляем рекурсивно:

$$\beta_i(T) = 1;$$

$$\beta_i(t) = \sum_{j=1}^N \beta_j(t+1) \cdot a_{ij} \cdot b_{ij}(O_{t+1}).$$

Процедура обновления переменных:

Определим временные переменные:

$$\gamma_i(t) = p(S_t = i | O, MAU) = \frac{\alpha_i(t) \cdot \beta_i(t)}{\sum_{j=1}^N \alpha_j(t) \beta_j(t)}$$

$$\xi_{ij}(t) = p(S_t = i, S_{t+1} = j | O, MAU) =$$

$$= \frac{\alpha_i(t) \cdot a_{ij} \cdot \beta_j(t+1) \cdot b_j(o_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t) \cdot a_{ij} \cdot \beta_j(t+1) \cdot b_j(o_{t+1})}$$

Имея γ и ξ , определим:

$$\bar{\pi}_i = \gamma_i(1), \quad \bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}, \quad \bar{b}_i(k) = \frac{\sum_{t=1}^T \delta_{o_t, o_k} \cdot \gamma_i(t)}{\sum_{t=1}^T \gamma_i(t)}.$$

Используя новые переменные A , B , π итерации

продолжаются до схождения.

В практической реализации как критерий останова данного алгоритма чаще всего применяют количество итераций или допустимую ошибку схождения.

Заключение

Модель активности пользователя основана на скрытой модели Маркова, в которой наблюдаемые переменные – это активность пользователя, а скрытый слой содержит состояния, которые измеряются интервалом времени до завершения работы с персональным компьютером. Метод Баума–Уэлча как способ идентификации параметров модели обеспечивает поиск локального максимума правдоподобия модели и имеющихся видимых наблюдений. Выбор начального распределения вероятностей на основе априорных знаний о процессе повышает точность данного метода.

Модель активности пользователя с модифицированным методом Баума–Уэлча может применяться как основа динамического управления питанием персональных компьютеров. Перспективой дальнейших исследований является экспериментальная оценка эффективности применения метода Баума–Уэлча в задаче параметрической идентификации модели активности пользователя.

Литература

1 Benini, L. *Dynamic Power Management: Design Techniques and CAD Tools*. [Text] / L. Benini, G. De Micheli. – Boston : Kluwer, 1998. – 248 с.

2 Snowdon, D. *Power Management and Dynamic Voltage Scaling*. [Text] / D. Snowdon, S. Ruocco, G. Heiser // *Proceedings of the National ICT Australia and School of Computer Science and Engineering University of NSW*, 2005. – P. 15-19.

3 *Conserving battery power - Windows Help* [Электронный ресурс]. – Режим доступа: <http://windows.microsoft.com/en-us/windows/conserving-battery-power#1TC=windows> – 13.02.2016.

4 *Competitive randomized algorithms for no uniform problems* [Text] / A. R. Karlin, M. S. Manasse, L. A. McGeoch, S. Owicki. // *Algorithmica*. – 1994. – № 11. – С. 542–571.

5 Ramanathan, D. *System level online power management algorithms*. [Text] / D. Ramanathan, R. Gupta. // *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*. – 2000. – С. 606 – 611.

6 Feldman, J. *A Fast Maximum-Likelihood Decoder for Convolutional Codes*. [Text] / J. Feldman, I. Abou-Faycal, M. Frigo. // *Vehicular Technology Conference*. – 2002. – №56. – С. 371–375.

7 Rabiner, L. *A tutorial on hidden Markov models and selected applications in speech recognition*. [Text] / L. Rabiner // *Proceedings of the IEEE*. – 1989. – № 2. – С. 257-286.

8 Matarazzo, T. J. *STRIDE for Structural Identification Using Expectation Maximization: Iterative Output-Only Method for Modal Identification* [Text] / T. J. Matarazzo, S. N. Pakzad. // *Journal of Engineering Mechanics*. – 2016. – № 142. – P. 37–55.

References

1. Benini, L., De Micheli, G. *Dynamic Power Management: Design Techniques and CAD Tools*. – Boston, Kluwer Publ., 1998. 248 p.

2. Snowdon, D., Ruocco, S., Heiser, G. *Power Management and Dynamic Voltage Scaling*. *National ICT Australia and School of Computer Science and Engineering University of NSW*, 2005, pp. 15-19.

3. *Conserving battery power - Windows Help*. Available at: <http://windows.microsoft.com/en-us/windows/conserving-battery-power#1TC=windows> (accessed 13.02.2016).

4. Karlin, A., Manasse, M., McGeoch, L., Owicki, S. *Competitive randomized algorithms for no uniform problems*. *Algorithmica*, 1994, no. 11, pp. 542–571.

5. Ramanathan, D., Gupta, R. *System level online power management algorithms*. *Design, Automation and Test in Europe Conference and Exhibition*, 2000, pp. 606–611.

6. Feldman, J., Abou-Faycal, I., Frigo, M. *Fast Maximum-Likelihood Decoder for Convolutional*

Codes. *Vehicular Technology Conference*, 2002, no. 56, pp. 371–375.

7. Rabiner, L. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989, no. (2), pp. 257-286.

8. Matarazzo, T., Pakzad, S. STRIDE for Structural Identification Using Expectation Maximization: Iterative Output-Only Method for Modal Identification. *Journal of Engineering Mechanics*, 2016, no. 142, pp. 37-55.

Поступила в редакцію 10.03.2016, рассмотрена на редколлегии 14.03.2016

МОДЕЛЬ АКТИВНОСТІ КОРИСТУВАЧА В ПОЛІТИЦІ КЕРУВАННЯ ЖИВЛЕННЯМ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА

І. Б. Туркін, О. В. Вдовітченко

У статті проаналізовано існуючі методи управління живленням персонального комп'ютера. Розглянуто модель керування живленням персонального комп'ютера і її реалізації в сучасних операційних системах. Запропоновано модель активності користувача, засновану на прихованій Марківській моделі. Активність користувача розглядається як на синтаксичному, так і на семантичному рівні. Розроблено структуру моделі активності користувача. Запропоновано та обґрунтовано параметричну ідентифікацію моделі методом Баума-Уелча. Модель активності користувача дозволить адаптувати політику динамічної системи управління енергоживленням з урахуванням фактора користувача.

Ключові слова: енергоспоживання, управління живленням, користувач, політика енергоживлення, ACPI, приховані Марківські моделі, метод Баума-Велша.

THE MODEL OF USER'S ACTIVITY IN POLICY OF PERSONAL COMPUTER POWER MANAGEMENT

I. B. Turkin, A. V. Vdovitchenko

The existing methods of personal computer power management are analyzed in this article. The model of personal computer power management and its realization in modern operation systems is considered. The model of user's activity which is based on hidden Markov model is proposed. The user's activity is considered at the syntactic and semantic levels. The structure of the model of user's activity is developed. The parametric identification is offered and justified by the Baum–Welch algorithm. The model of user's activity gives the opportunity to adapt the policy of the dynamic system of the power consumption considering user's factor.

Key words: power consumption, power management, user, energy policy, ACPI, hidden Markov model, the Baum-Welch method.

Туркин Игорь Борисович – д-р техн. наук, профессор, зав. каф. інженерії програмного забезпечення, Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: turkinse@gmail.com

Вдовитченко Александр Валерьевич – аспірант каф. інженерії програмного забезпечення, Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: kentsanya91@gmail.com

Turkin Igor' Borisovich – Dr. Sc. in Engineering, Prof., Head of Dep. of Software Engineering, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: igor.turkin.khai@gmail.com.

Vdovitchenko Aleksandr Valer'evich – Postgraduate, Department of Software Engineering, National Aerospace University " Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: kentsanya91@ gmail.com.