Modelling Battery Behaviour Using Chipset Energy Benchmarking

R. Damasevicius[1], J. Toldinas[2], G. Grigaravicius[2]
[1]*Department of Software Engineering, Kaunas University of Technology, Studentu St. 50, LT-51368 Kaunas, Lithuania*
[2]*Department of Computer Science, Kaunas University of Technology, Studentu St. 50, LT-51367 Kaunas, Lithuania*
robertas.damasevicius@ktu.lt

***Abstract*—Despite advances in low power system design, short battery life remains a significant user concern. Effective management of energy resources available on a mobile device requires understanding of the principles of battery behaviour. We propose a time-delay model of a battery, which depends upon three non-linear processes: rate-capacity effect, recovery effect and software scheduling effect. We provide an analysis of the power consumption results using 3DMark'06 chipset benchmarks and demonstrate that a moderate-to-strong correlation between power consumption vs. CPU load, memory allocation and memory release is observed. Finally, we apply our model for chipset energy efficiency profiling and propose a power benchmark metric.**

***Index Terms*—Energy consumption modelling, battery behaviour, time-delay model, power benchmarking.**

I. INTRODUCTION

Short battery life was, and still remains, a significant concern for mobile device users. There is a big gap between the energy resources needed by the mobile device and the energy available from the battery; therefore, the average battery life of actively used mobile devices is usually less than two days [1]. Another matter of concern is the ability to predict battery life using available information on the application usage habits, the device's modes of operation and power management schemes so that the user could decide how to use the remaining battery time most effectively. Such prediction is only possible when the behaviour of the battery can be modelled accurately taking all internal (electro-chemical) and external (CPU load, memory usage, display rendering, etc.) factors that influence the *State of Charge* (SoC) of the battery into account.

The most important factors influencing its lifetime are the battery's capacity and the battery's discharge rate. This rate is influenced by three non-linear processes, two of which are determined by the electrochemical properties of the battery. The *rate-capacity effect* is observed when a battery is discharged continuously; a high discharge current causes a battery to provide less energy until the end of its lifetime as compared to a lower discharge current [2]. The *recovery effect* causes the battery capacity to recover to a certain extent during periods of low or no discharge. A third non-linear process that has influence on the state of the battery is *software scheduling* schemes introduced at application [3] or operating system level, which control, e.g., CPU rate and energy consumption level of peripheral devices that are considered non-essential for some applications such as display brightness. The result of these effects is the dependency of the battery lifetime upon battery discharge distribution over time [4], which in turn depends upon user behaviour and usage patterns. Furthermore, the peak power usage can sometimes be a more important factor in determining battery capacity than average power usage [2]. Finally, the effective capacity of a battery depends on the rate at which it is discharged, because the electrochemical actions in the batter take a finite time to complete and they cannot follow the battery load instantaneously.

To investigate the influence of the device workload on the battery lifetime, a battery model is needed that includes the above described effects. In electrical engineering, electrical circuit models [5], [6], and electro-chemical models [7] are used. Also high-level analytic and stochastic battery models [4] are available. We treat the battery, CPU and memory data gathered during the execution of the computer benchmarking tests as complex time series. An accurate identification of the dynamics underlying complex time series, is of crucial importance in understanding the corresponding physical process, and in turn affects the subsequent model development [8].

We propose to use time-delay models to study the relationships between CPU load, physical memory usage (allocation, release) and battery charge levels. Time-delay models have been used previously to explain many natural, biological and social processes such as pray-predator systems [9], precipitation patterns [10], business cycles [11].

II. TIME-DELAY MODEL OF BATTERY'S STATE OF CHARGE (SOC)

The nonlinear battery system can be modelled by a state equation and an output equation [6]:

$$\begin{cases} x_{k+1} = f(x_k, u_k) + w_k, \\ y_k = g(x_k, u_k) + v_k, \end{cases} \quad (1)$$

Manuscript received December 20, 2013; accepted April 2, 2013.

where x is the system state, u is the system's input, w is the unmeasured process noise that affects the system's state, y is the output of the system, v is the measurement noise, and k is the discrete time index of the time series.

The state of the battery can be described by the following time delay differential equation [12]

$$\dot{x}(t) = f(x(t), x(t-\tau)), \quad (2)$$

where τ is the process lag.

In constructing the battery behaviour model, we use the following assumptions:

1) The state of a battery is determined by two factors: CPU load and physical memory usage.
2) Due to slow electrochemical process, the change of the battery charge level lags behind the changes in CPU usage, memory allocate and release events.

We formulate our time-delay model of the battery's state of charge (SoC) as follows

$$\dot{c}(t) = f(c(t), c(t-\tau_c), l(t-\tau_l), m_a(t-\tau_a), m_r(t-\tau_r)), \quad (3)$$

where $c(t)$ is the battery's SoC, $l(t)$ is the CPU load process, $m_a(t)$ is the physical memory access process, $m_r(t)$ is the physical memory release process, τ_c is the lag of the battery discharge process, τ_l is the lag of the CPU load process, τ_a is the lag of the memory access process, and τ_r is the lag of the memory release process.

Battery discharge is a continuous process while the events that draw power (CPU calls, memory access and release events) are discrete, and the dependent variable (measured battery charge level) is also discrete. Moreover, the value of CPU load is instantaneous, while the number of memory accesses is only known over a time span. To represent these values as the continuous ones, some smoothing is required.

Let l_t be the CPU load value (in percents) in time t. Let m_t be the amount of used physical memory in time t. Let c_t be the battery charge value (in percents) in time t. Let $h(x)$ be a step function $h(x) = \{^{x,\ x \geq 0}_{0,\ x < 0}$. Physical memory accessed in a time span $(t_0, t_0 + \Delta t)$ is estimated as $m_a(t_0, \Delta t) = \sum_{t=t_0}^{t_0+\Delta t} h(m_t - m_{t-1})$. Physical memory released in a time span $(t_0, t_0 + \Delta t)$ is estimated as $m_r(t_0, \Delta t) = \sum_{t=t_0}^{t_0+\Delta t} h(m_{t-1} - m_t)$.

Let $f_s(X, t_0, w) = \frac{1}{w}\sum_{t=t_0}^{t_0+w-1} x_t$ be the smoothing function (we use moving average, MVA) of the time series $x_t \in X$, and w is the length of a smoothing window. Averaged CPU load over window w_l is $\overline{l(t)} = f_s(l_t, t_0, w_l)$. Averaged physical memory access over window w_a is $\overline{m_a(t)} = f_s(m_a, t_0, w_a)$. Averaged physical memory release over window w_r is $\overline{m_r(t)} = f_s(m_r, t_0, w_r)$. Averaged battery charge value over window w_c is $\overline{c(t)} = f_s(c_t, t_0, w_c)$.

To analyse the model, we formulate the following hypotheses:

B. Hypothesis H1

There is no time-delay relationship between the battery charge level and CPU load.

Let $\overline{c(t+\tau)} \sim g(\overline{l(t)})$ be a hypothetical functional relationship between $\overline{c(t)}$ and $\overline{l(t)}$, where τ is the lag value. Pearson correlation of $\overline{c(t)}$ and $\overline{l(t)}$ is $\rho_{c,l} = \max_{\tau, w_c, w_l} \rho(\overline{c(t+\tau)}, \overline{l(t)})$, where w_c and w_l are smoothing parameters of $\overline{c(t)}$ and $\overline{l(t)}$.

C. Hypothesis H2

There is no time-delay relationship between the battery charge level and memory access events.

Let $\overline{c(t+\tau)} \sim g(\overline{m_a(t)})$ be a hypothetical functional relationship between $\overline{c(t)}$ and $\overline{m_a(t)}$. Pearson correlation of $\overline{c(t)}$ and $\overline{m_a(t)}$ is $\rho_{c,m_a} = \max_{\tau, w_c, w_a} \rho(\overline{c(t+\tau)}, \overline{m_a(t)})$, where w_c and w_a are smoothing parameters of $\overline{c(t)}$ and $\overline{m_a(t)}$.

D. Hypothesis H3

There is no time-delay relationship between the battery charge level and memory release events.

Let $\overline{c(t+\tau)} \sim g(\overline{m_r(t)})$ be a hypothetical functional relationship between $\overline{c(t)}$ and $\overline{m_r(t)}$. Pearson correlation of $\overline{c(t)}$ and $\overline{m_r(t)}$ is $\rho_{c,m_r} = \max_{\tau, w_c, w_r} \rho(\overline{c(t+\tau)}, \overline{m_r(t)})$, where w_c and w_a are smoothing parameters of $\overline{c(t)}$ and $\overline{m_r(t)}$.

III. Application of the Time-delay Model for Power Benchmarking

Energy efficiency is a critical design factor on a battery-powered mobile device. Typically, computer benchmarks are used to evaluate the performance of a computer by performing a strictly defined set of operations and returning some numerical result (a metric) describing how well the tested computer performed. Running the same benchmark test on multiple computers allows the computers to be compared with respect to their performance.

To evaluate energy efficiency of a device, power benchmarks [13] are used. Known examples of such benchmarks include SPECpower ssj2008 [14] benchmark for measuring the performance and energy consumption of a system running Java-based workloads, TPC-Energy [15] metric and its extensions such as the dynamic weighted energy-efficiency benchmark (DWEE) [16].

Here we propose a power benchmarking metric based on the time-delay model of the battery's SoC as follows (the smaller value of the metric is the better one)

$$P_b = \frac{s_c}{s_{perf}} \bullet \left(c(t_{start}) - \frac{\sum_{t=t_{start}}^{t_{end}} c(t)}{\Delta t(t_{end} - t_{start})} \right), \quad (4)$$

where t_{start} is start time of benchmark run, t_{end} is time at the end of benchmark run, Δt is sampling period, s_{perf} is the value of the benchmark performance score, and s_c is a scaling constant introduced for usability purposes and is equal to 1000, if $c(t)$ is evaluated in percents.

IV. Case Study and Experimental Results

We use the Futuremark® 3DMark®06 version 1.2.0 benchmark, which provides tests for testing the DirectX 9 gaming performance of Windows PCs and devices, graphics, CPU and GPU feature tests. We registered battery charge level, CPU load and free physical memory every 1 s starting from the fully charged battery. The experiments were performed on Hewlett-Packard F.23 laptop PC running Windows 7 Ultimate 32 bit OS on Intel® Core Duo T2250 1.73 GHz CPU, 4GB DDR2 RAM 265.4 MHz, Mobile Intel® 945 Express Chipset, i945GM 400 MHz GPU with 8MB internal DDR2 memory. We used the measurement methodology already described in [3]. The results of measurements are presented in Figs. 1-3. Fig.1 shows the battery charge level, Fig. 2 shows CPU load and Fig. 3 shows the free physical memory in a computer registered during 9 consecutive runs of the standard 3DMark®06 test.

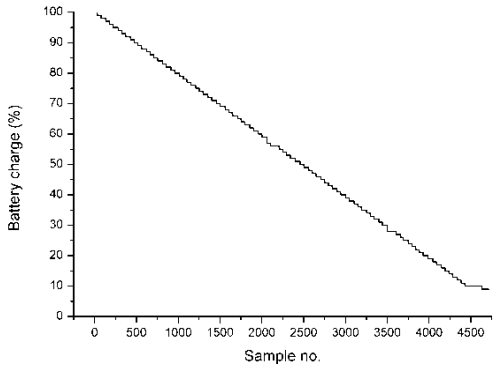

Fig. 1. Battery charge level during a standard 3DMark®06 test.

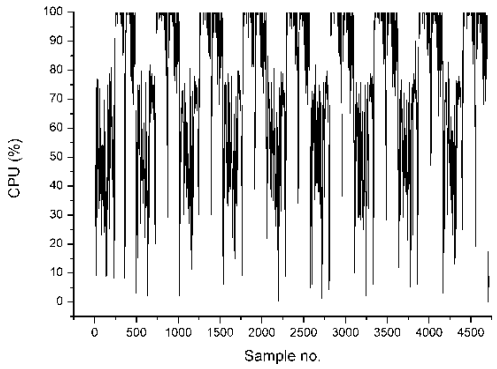

Fig. 2. CPU load during a standard 3DMark®06 test.

The time-delay model of the battery's SoC is evaluated in Fig. 4-6. Fig. 4 shows the relationship of lagged (lag = 340 s) MVA (window length is 60 s) of the CPU load value vs. MVA (window length is 150 s) of power consumption. The lag value corresponds to the largest value of the Pearson correlation (see Table I). The battery has two distinct states: 1) high-load (CPU load > 70%), mean MVA of consumed power is 1.20%; 2) low-load (CPU load < 70%), mean MVA of consumed power is 1.07%.

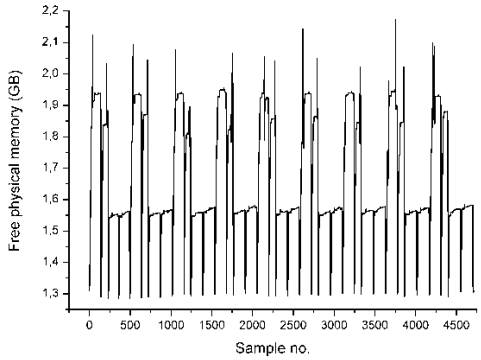

Fig. 3. Physical memory use during a standard 3DMark®06 test.

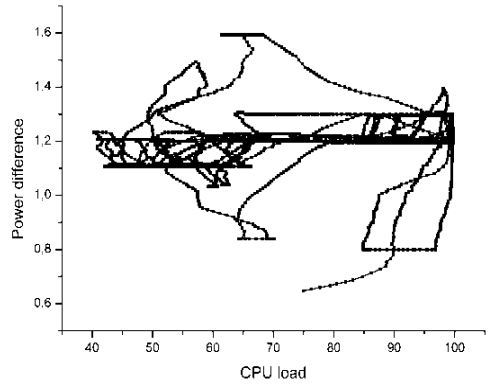

Fig. 4. MVA of CPU load vs. MVA of power consumption.

Fig. 5 shows the relationship of the lagged (lag = 520 s) MVA (window length is 180 s) of the memory allocation value vs. MVA (window length is 150 s) of the power consumption. The plot demonstrates the complex multi-state behaviour of power consumption process for high (> 256 MB) memory allocation events.

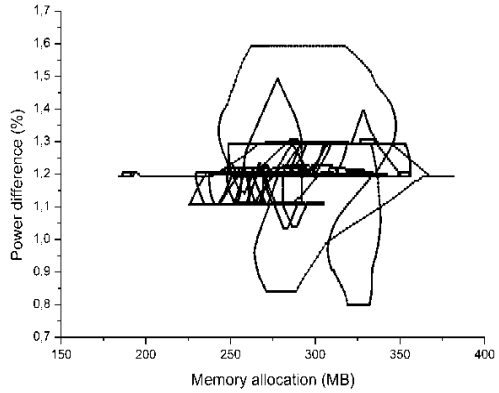

Fig. 5. MVA of physical memory allocation vs. MVA of power consumption.

Fig. 6 shows the relationship of the lagged (lag = 180 s) moving average (window length is 480 s) of the memory allocation value vs. moving average (window length is 150 s) of the power consumption. The plot is similar to Fig. 5: for high (> 256 MB) memory release events, power

consumption becomes less predictable.

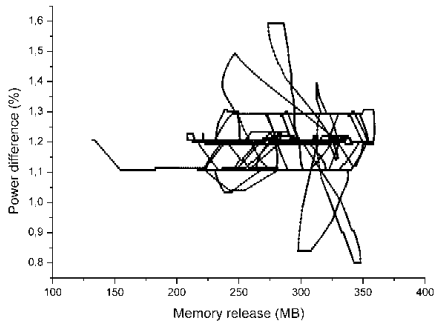

Fig. 6. MVA of physical memory release vs. MVA of power consumption.

We evaluate the hypotheses using the two-tailed test of the Pearson correlation coefficients. The correlations coefficients are significant at p = 0.001 (ρ > 0.104 for N = 1000). The relationship is usually considered strong, if ρ > 0.4, and moderate, if ρ > 0.3. The results are summarized in Table I. Based on these results, we reject the hypotheses H1, H2 and H3, and confirm the dependency of the battery's charge level upon time-delayed values of CPU load and memory access and release values.

TABLE I. PEARSON CORRELATION OF MVAS OF CPU LOAD, PHYSICAL MEMORY ALLOCATION AND RELEASE VS. MVA OF POWER CONSUMPTION.

Process	MVA window size, s	Lag value, s	Pearson correlation	Hypothesis outcome
CPU load	60	340	0,603	Rejected
Memory allocation	150	520	0,360	Rejected
Memory release	480	180	0,590	Rejected

To validate the power benchmark metric, we have run the standard 3DMark®06 test 8 times on the HP laptop PC and have averaged the results. We have obtained the 3DMark®06 score value of 224 and the power metric value (Eq. 4) P_B = 24.0 (std. deviation = 1.5).

To compare, we have run the benchmark on the Acer Aspire 1800 laptop PC running Windows 7 Professional 32 bit OS on Intel® Pentium 4 2.93 GHz CPU, 1GB DDR RAM 166MHz, Intel® i915P/i915g chipset, and PA3206U (59Ah, 17V) battery and have obtained the 3DMark®06 test score value of 138 and the power metric value P_B = 303.9 (std. deviation = 18.7).

Based on these results we conclude that the Acer Aspire has worse energy efficiency due to higher CPU speed that requires more energy, smaller RAM (which means more physical memory accesses are required) and worse chipset (including GPU) characteristics.

V. CONCLUSIONS

1) Battery is a time-delay system with the experimentally-determined lag values of CPU is 340 s, physical memory allocation lag is 340 s and release lag is 180 s.

2) There is no strong correlation between CPU load and power consumption, however there is a strong correlation between the memory usage and power consumption which corresponds well to long-known observation that memory is the largest consumer of power in modern computers thus confirming the validity of the proposed model.

3) Lag between energy consumption events and battery charge value decreases the predictability of the battery's State of Charge (SoC).

4) Battery-powered computers have the properties of multi-state systems with complex relations between states therefore battery charge forecasting is difficult.

5) The proposed time-delay model of the battery's SoC can be used as a theoretical background for power efficiency benchmarking.

6) We propose a formula for computing the power efficiency metric based on the 3DMark®06 test score and the battery charge level measurement results.

REFERENCES

[1] N. Korhonen, "Predicting mobile device battery life", M.S. thesis, Aalto University, Finland, 2011.

[2] T. L. Martin, "Balancing Batteries, Power, and Performance: System Issues in CPU Speed-Setting for Mobile Computing", Ph.D. dissertation, Carnegie Mellon University, USA, 1999.

[3] J. Toldinas, V. Stuikys, R. Damasevicius, G. Ziberkas, "Application-level energy consumption in communication models for handhelds", *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, no. 6, pp. 73–76, 2009.

[4] M. R. Jongerden, "Model-based energy analysis of battery powered systems", Ph.D. dissertation, University of Twente, the Netherlands, 2010.

[5] S. Gold, "A PSPICE macromodel for lithium-ion batteries", in *Proc. of the 12th Annual Battery Conference on Applications and Advances*, Long Beach, CA, USA, 1997, pp. 215–222.

[6] F. Xuyun; S. Zechang; "A battery model including hysteresis for State-of-Charge estimation in Ni-MH battery", in *Proc. of the Vehicle Power and Propulsion Conference (VPPC 08),* 2008, pp. 1–5.

[7] M. Doyle, T. F. Fuller, J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell", *Journal of the Electrochemical Society*, vol. 140, no. 6, pp. 1526–1533, 1993. [Online]. Available: http://dx.doi.org/10.1149/1.2221597

[8] K. Pukenas, K. Muckus, "Algorithm for Detecting Deterministic Chaos in Pseudoperiodic Time Series", *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, no. 8, pp. 53–56, 2007.

[9] Y. Qu, J. Wei, "Bifurcation analysis in a time-delay model for prey-predator growth with stage-structure", *Nonlinear Dynamics*, vol. 49, pp. 285–294, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11071-006-9133-x

[10] R. B. Smith, "A linear upslope-time- delay model for orographic precipitation", *Journal of Hydrology*, vol. 282, no. 1–4, 2003.

[11] L. Zhou, Y. Li, "A dynamic IS-LM business cycle model with two time delays in capital accumulation equation", *Journal of Computational and Applied Mathematics,* vol. 228, no. 1, pp. 182–187, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cam.2008.09.004

[12] J. E. Forde, "Delay Differential Equation Models in Mathematical Biology", Ph.D. dissertation, University of Michigan, USA, 2005.

[13] M. Marcu, D. Tudor, S. Fuicu, S. Copil-Crisan, F. Maticu, M. Micea, "Power efficiency study of multi-threading applications for multi-core mobile systems", *WSEAS Transactions on Computers*, vol. 7, no. 12, pp. 1875–1885, 2008.

[14] *Standard Performance Evaluation Corporation (SPEC), SPECpower ssj2008,* 2008. [Online]. Available: http://www.spec.org/power ssj2008

[15] M. Poess, R. Othayoth Nambiar, K. Vaid, J. M. Stephens, Jr., Karl Huppler, and E. Haines, "Energy benchmarks: a detailed analysis", in *Proc. of the 1st Int. Conference on Energy-Efficient Computing and Networking (e-Energy 10)*, ACM, New York, USA, 2010, pp. 131–140.

[16] D. Schall, V. Hoefner, and M. Kern, "Towards an enhanced benchmark advocating energy-efficient systems", in *Proc. of the Third TPC Technology conference on Topics in Performance Evaluation, Measurement and Characterization (TPCTC 11)*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 31–45.