Energy Minimization via Dynamic Voltage Scaling for Real-Time Video Encoding on Mobile Devices

Ming Yang, Yonggang Wen, Jianfei Cai and Chuan Heng Foh
School of Computer Engineering, Nanyang Technological University, Singapore 639798
Email: {yang0258, ygwen, asjfcai, aschfoh}@ntu.edu.sg

***Abstract*—This paper investigates the problem of minimizing energy consumption for real-time video encoding on mobile devices, by dynamically configuring the clock frequency in the CPU via the dynamic voltage scaling (DVS) technology. The problem can be formulated as a constrained optimization problem, whose objective is to minimize the total energy consumption of encoding video contents while respecting a real-time delay constraint. Under a probabilistic workload model, we obtain closed-form solutions for both the optimal clock frequency configuration and the resulted minimum energy. We also compare the optimal solution with a brute force flat frequency configuration. Numerical results indicate that our derived optimal solution outperforms the brute-force approach significantly. Moreover, we apply the optimal solution for real-time H.264/AVC video encoding application. Our numerical results suggest that an energy saving of $10\% - 20\%$ can be achieved, compared to the flat clock frequency scheduling.**

I. Introduction

Growing popularity of smart phones and ubiquitous wireless Internet access have fueled an exponential growth of mobile media [1]. Many mobile devices (e.g., iPhone) nowadays are capable of capturing high-quality photos and videos [2]. These user-generated contents are encoded by mobile devices first and then uploaded to the cloud via wireless connections or portable storage devices (e.g., Compact Flash Card). Such an emerging media trend is contributing significantly to the growth of mobile data traffic, which is expected to increase by a factor of 40 between 2009 and 2014 [3]. In particular, by 2015 video traffic will constitute almost two-thirds of the world's mobile data traffic.

However, mobile devices are inherently resource-constrained [4]. In particular, the energy supply on mobile devices is limited by the physical size of the battery that cannot grow in response to high demand. As a result, the limited battery life-time has been shown to be the most important factor affecting the user experience [5]. The emerging trend of video encoding on mobile devices, due to its energy-hungry nature, aggravates this limitation. In order to sustain a longer battery life-time, the task of video encoding on mobile devices should be executed with energy concern [6]. In this research, we aim to minimize the energy consumption for video encoding on mobile devices, while respecting some quality-of-service (QoS) requirements.

Previous researchers have investigated the problem of energy-aware video encoding from a perspective of encoder design. In [7], a comprehensive power-rate-distortion model was developed to describe the relationship of different encoding modules for the general video coding structure. In [8], a joint complexity-distortion optimization approach was proposed for real-time H.264 video encoding under power-constraint, in which computational resource is dynamically allocated to frames and Macro-Blocks. The proposed system needs to dynamically allocate resource to motion estimation and mode decision modules and configure the two modules to utilize the resource. However, these encoder-centric approaches often resulted in algorithms that are complicated, rendering its applicability to resource-poor mobile devices to a limited level.

In this research, we propose an alterative venue of minimizing the energy consumed for video encoding on mobile devices by dynamically reconfiguring the clock frequency of the chip. Our proposed solution is feasible due to the dynamic voltage scaling technology (DVS). In CMOS circuits[9], the energy per operation $\mathcal{E}_{op}$ is proportional to V^2, where V is the supply voltage to the chip. Moreover, it has been observed that the clock frequency of the chip, f, is approximately linearly proportional to the voltage supply of V[9]. Therefore, the energy per operation can be expressed as, $\mathcal{E}_{op} = \kappa f^2$, where κ is the energy coefficient depending on the chip architecture. Note that CPU can reduce its energy consumption substantially by running more slowly [10]. However, for real-time video encoding, the encoder has to meet a delay deadline for each group of pictures (GOP), which suggests that the clock frequency cannot be constantly small.

In this paper, we take a systematic approach to investigate the problem of how to dynamically reconfigure the clock frequency in the mobile device to minimize the energy consumption, while respecting the QoS requirement. We adopt a probabilistic QoS model, in which the encoding process should complete with a target probability within a specified delay deadline for each GOP. Such a requirement is translated into the number of CPU cycles required before the encoding deadline. Under this model, the optimal clock-scheduling problem is formulated as a constrained optimization problem, in which the objective is to minimize the total energy consumption with a constraint of delay deadline. We solve the optimization problem analytically and obtain closed-form solutions for both the optimal clock frequency schedule and the minimum energy consumption. We then apply the lightweight clock scheduling algorithm to real-time H.264/AVC encoding application on mobile devices. The numerical results suggest that significant amount of energy can be saved by using our optimal solution.

The rest of the paper is organized as follows. In Section II, we present a mathematical model for energy consumption in mobile devices and encoder workload, and a problem formulation for optimal clock scheduling mechanism. In Section III, we solve the optimization problem and obtain closed-form solutions for the optimal clock-scheduling algorithm and the minimum energy consumption. In Section IV, the lightweight algorithm is applied for real-time H.264/AVC encoding on mobile devices and we obtain numerical results about the energy saving. Section V concludes this paper.

II. Model and Formulation

In this section, we first present a mathematical model for energy consumption in mobile devices and a probabilistic model for encoder workload. Under this model, the problem of optimal clock-scheduling mechanism is formulated as a constrained optimization problem.

A. Energy Consumption Model for Mobile Devices

The energy consumed on mobile devices, for a special computing task, depends on the number of CPU cycles and the clock frequency. First, in CMOS circuits, the clock frequency f, is approximately linearly proportional to the voltage supply V, and the energy per cycle E_c is proportional to V^2 [9]. Therefore, the energy consumption per cycle can be expressed as

$$E_c = \kappa f^2, \tag{1}$$

where κ is a coefficient depending on the chip architecture. The energy per CPU cyle, as denoted in (1), has the following properties including:

- $E_c(f)$ is an increasing function of the clock frequency of f;
- $E_c(f)$ is a convex function of the clock frequency of f.

Given these properties, it can be seen that CPU can conserve energy substantially by running more slowly. However, for real-time video encoding, the encoder has to meet a specified deadline for each GOP, which suggests that the clock frequency cannot be constantly small. Therefore properly scheduling of CPU clock frequency can conserve energy while meeting the required deadline simultaneously.

B. Probabilistic Workload Model for Video Encoding

The workload of an encoding task is characterized by the number of CPU cycles, denoted as W. It is normally modeled as a random variable. As shown in [11] [12], the (truncated) normal distribution can be used to model the workload. The probability density function (PDF) of normal distribution is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad \text{for} \quad x > 0. \tag{2}$$

In this paper, we assume a probabilistic workload model that an encoding task should be completed with probability ρ by allocating W_ρ cycles. This requirement can be expressed as the cumulative distribution function (CDF)

$$F(x) = \text{Pr}[x \leq W_\rho] \geq \rho. \tag{3}$$

As such, the required number of CPU cycles W_ρ, for an empirical normal distribution and completion probability ρ, is given by

$$W_\rho = F_W^{-1}(\rho). \tag{4}$$

In this model, a CPU frequency scheduling consists of two parts, including the pre-deadline part and the post-deadline part. The maximum number of cycles executed in the pre-deadline part equals to W_ρ. The post-deadline part describes the scheduling when task has missed its deadline. In this paper, we only focus on the scheduling policy for the pre-deadline part. If the encoding process misses its required deadline, it is assumed that the post-deadline part is executed with a maximum clock frequency.

C. Problem Formulation

In the real-time video encoding system, we focus on the encoding task of each GOP. Specifically, the encoding task of each GOP is required to meet the deadline at a specified probability, which can be expressed as a probabilistic QoS model,

$$\text{Pr}[t \leq T] \geq \rho, \tag{5}$$

where t is the encoding time for an individual GOP-encoding task and T is the required encoding deadline for a GOP-encoding task.

The encoding requirement specified in Eq.(5), under the probabilistic workload model, can be translated into a requirement of the number of CPU cycles, i.e. W_ρ defined in Eq.(4). Therefore, the total energy consumption can be derived as

$$\begin{aligned} \varepsilon_c &= \kappa \int_0^{W_\rho} p(x) \int_0^x [f(w)]^2 dw dx \\ &\overset{(a)}{=} \kappa \int_0^{W_\rho} [f(w)]^2 \int_w^{W_\rho} p(x) dx dw \\ &\overset{(b)}{=} \kappa \int_0^{W_\rho} [f(w)]^2 (1 - F(w)) dw, \end{aligned}$$

where $f(w)$ is the clock frequency defined as a function of w, which is the number of CPU cycles that has been completed for the current task, (a) results from the exchange of integral order, and (b) is from the definition of the CDF.

A discrete version of the energy consumption, by using the approximation of $du = 1$, can be written as

$$\varepsilon_c = \kappa \sum_{w=1}^{W_\rho} F^c(w) [f(w)]^2, \tag{6}$$

where $F^c(w)$ is the complementary cumulative distribution function (CCDF) of workload. Notice that $F^c(w)$ is the probability in which the encoding task has not finished after executing w CPU cycles.

Using the above definition, we can formulate the optimal clock frequency allocation problem as the following constrained optimization problem,

$$\min_{f(w)} \quad \varepsilon_c = \kappa \sum_{w=1}^{W_\rho} F^c(w)[f(w)]^2, \tag{7}$$

$$s.t. \quad \sum_{w=1}^{W_\rho} \frac{1}{f(w)} \leq T, \tag{8}$$

$$f(w) > 0,$$

where the constraint of Eq.(8) corresponds to the task deadline requirement.

III. Optimal DVS scheduling

In this section, we solve the optimization problem via a Lagrangian method and obtain the closed-form solutions for the optimal clock-scheduling policy and the corresponding minimum energy consumption. We then evaluate the characteristics of the clock-scheduling policy.

A. Derivation of Optimal DVS policy

In this subsection, we first show the existence of a unique solution to the aforementioned optimization problem and then use a Lagrangian multiplier method to solve the optimization problem in Eq.(7).

First, we show the existence of a unique solution for the optimization problem. In Eq.(7), the energy expression is a linear combination of $[f(w)]^2$. Since the form of x^2 is a convex function, our objective function Eq.(7) is also a convex function. It is clear that the two constrains in Eq.(8) are both convex sets [13]. Therefore, we can conclude that there exists a unique solution for the convex optimization problem.

Second, we use the Lagrangian multiplier method to solve the optimization problem in Eq.(7). The Lagrangian function is given by

$$\begin{aligned} L(f(w), \lambda) &= \sum_{w=1}^{W_\rho} F^c(w)[f(w)]^2 + \lambda(\sum_{w=1}^{W_\rho} \frac{1}{f(w)} - T) \\ &= \sum_{w=1}^{W_\rho} \{F^c(w)[f(w)]^2 + \frac{\lambda}{f(w)}\} - \lambda T. \end{aligned}$$

Using KKT condition, the optimization problem must satisfy the following conditions,

$$\frac{\partial L(f(w), \lambda)}{\partial f(w)} = 2F^c(w)f(w) - \frac{\lambda}{[f(w)]^2} = 0 \tag{9}$$

$$\frac{\partial L(f(w), \lambda)}{\partial \lambda} = \sum_{w=1}^{W_\rho} \frac{1}{f(w)} - T = 0. \tag{10}$$

From Eq.(9) we can obtain

$$f^*(w) = \{\frac{\lambda}{2F^c(w)}\}^{1/3}. \tag{11}$$

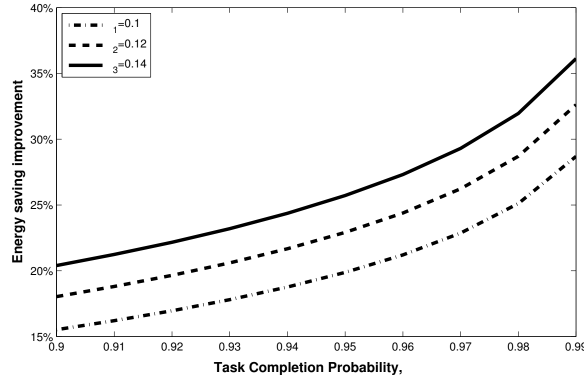


Fig. 1. The energy saving improvement compared to flat scheduling. $\eta = \sigma/\mu$, and η_1, η_2, η_3 are respectively $0.1, 0.12, 0.14$.

Substituting Eq.(11) into Eq.(10), we can obtain

$$(\frac{\lambda}{2})^{1/3} = \frac{\sum_{w=1}^{W_\rho}[F^c(w)]^{1/3}}{T}. \tag{12}$$

Therefore, substituting Eq.(12) into Eq.(11), the optimal CPU frequency scheduling policy is given by

$$f^*(w) = \frac{\theta}{T[F^c(w)]^{1/3}}, \tag{13}$$

where

$$\theta = \sum_{i=1}^{W_\rho}[F^c(i)]^{1/3}. \tag{14}$$

Substituting Eq.(13) into Eq.(6), we obtain the expected optimal energy consumption as

$$\begin{aligned} \varepsilon_c^* &= \frac{\kappa}{T^2}\{\sum_{i=1}^{W_\rho}[F^c(i)]^{1/3}\}^3 \\ &= \frac{\kappa}{T^2}\theta^3. \end{aligned} \tag{15}$$

In this research, we also consider a benchmark scheduling policy, i.e. a brute force approach that adopts a flat frequency scheduling. The brute force approach has the same amount of pre-deadline workload with our proposed optimal DVS scheduling. The lowest frequency for the flat frequency scheduling scheme is

$$f^F(w) = W_\rho/T. \tag{16}$$

In this case, the minimum energy consumption is

$$\mathcal{E}_c^F = \kappa W_\rho^3/T^2. \tag{17}$$

It will be shown in the next section, for a probabilistic workload, our optimal DVS frequency allocation can conserve considerable energy most time, compared to the flat frequency scheduling.

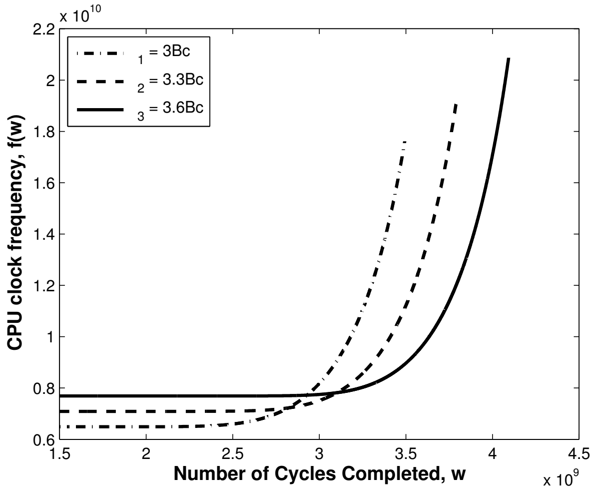


Fig. 2. The optimal clock frequency scheduling. The mean of each workload are respectively $\mu_1 = 3Bc$, $\mu_2 = 3.3Bc$, $\mu_3 = 3.6Bc$, while standard derivation is fixed to $\sigma = 0.3Bc$, and deadline is $T = 0.5s$. (Bc denotes Billion cycle)

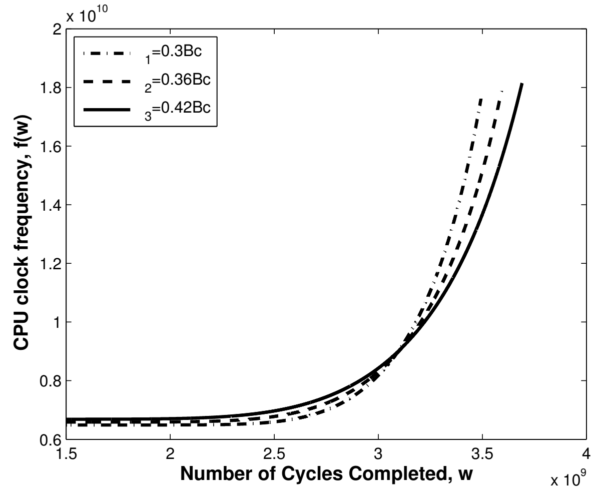


Fig. 3. The optimal clock frequency scheduling. The standard derivation of each workload are respectively $\sigma_1 = 0.3Bc$, $\sigma_2 = 0.36Bc$, $\sigma_3 = 0.42Bc$, while mean of cycles is fixed to $\mu = 3Bc$, and deadline is $T = 0.5s$.

B. Optimal DVS Scheduling Characteristics

In this subsection, we investigate the characteristics of the optimization solution, including the energy saving for different workloads and the optimal DVS scheduling relationship between different workloads.

First, let us consider the energy saving performance. We need to know the potential improvement capability of energy saving under different workload (e.g., μ, σ combinations). Compared to the flat frequency scheduling policy, the energy saving of our proposed optimal DVS scheduling policy, denoted as

$$\delta = \frac{\mathcal{E}_c^* - \mathcal{E}_c^F}{\mathcal{E}_c^F}, \tag{18}$$

is plotted in Figure 1, as a function of the task completion probabilty, for different variance-to-mean ratios ($\eta = \sigma/\mu$). We can see that the energy saving increases with the increasing of the task completion probability. Moreover for larger variance-to-mean ratio, we can obtain more energy saving, which means higher variance of workload can potentially result in more energy saving.

Second, let us consider the optimal clock-frequency policy. In Figure 2, we compare the optimal clock-frequency policies for three different means of the workload cycles (μ), with the same variance of the workload cycles (σ^2). We observe that the shape of frequency scheduling curves for different μ is similar to each other, as shown analytically next.

Let us consider two workloads of WL_1, WL_2, with $\mu_1 > \mu_2$ and $\sigma_1 = \sigma_2$. In normal distributions, the PDF curve of WL_1 can be obtained by right shifting the WL_2's curve with a distance of $\Delta\mu = \mu_1 - \mu_2$. The CDF and CCDF curves of them follow the same shifting rule. As a result, we can get the relationship,

$$F_2^c(w) = F_1^c(w + \Delta\mu). \tag{19}$$

Since the shapes of CCDF for two distribution have a shifting relationship, the difference between θ can be derived as follows,

$$\begin{aligned} \theta_1 - \theta_2 &= \sum_{i=1}^{W_{\rho_1}} [F_1^c(i)]^{1/3} - \sum_{i=1}^{W_{\rho_2}} [F_2^c(i)]^{1/3} \\ &= \Delta\mu = \mu_1 - \mu_2. \end{aligned} \tag{20}$$

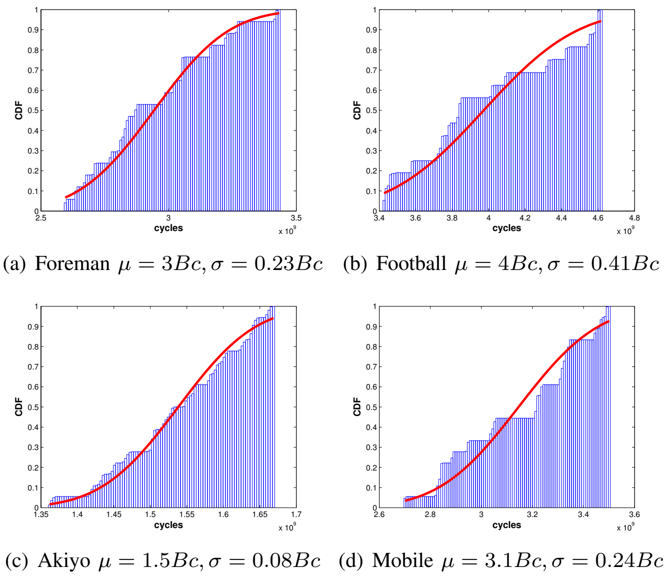
Using this result, we can obtain the following relationship between their corresponding clock-frequency policies,

$$\begin{aligned} f_2^*(w) &= \frac{\theta_2}{T[F_2^c(w)]^{1/3}} \\ &= \frac{\theta_2}{T[F_1^c(w+\Delta\mu)]^{1/3}} \\ &= \frac{\theta_1 - \Delta\mu}{\theta_1} f_1^*(w + \Delta\mu). \end{aligned} \tag{21}$$

Therefore, the optimal frequency scheduling vector of WL_2 can be calculated as a left shift $\Delta\mu$ with a scale of $\frac{\theta_1 - \Delta\mu}{\theta_1}$ from the scheduling vector of WL_1.

Using the shift-and-scale property, we can reduce the complexity of our proposed optimal DVS scheduling algorithm significantly. Specifically, for a given work load distribution, the optimal DVS scheduling policy, as denoted in (13), can be stored in a table. For practical encoding applications, the practical clock frequency scheduling vector can be obtained by the shift-and-scale approach. Therefore, the shift-and-scale relationship could be used to simplify the frequency optimization algorithm on practical platforms.

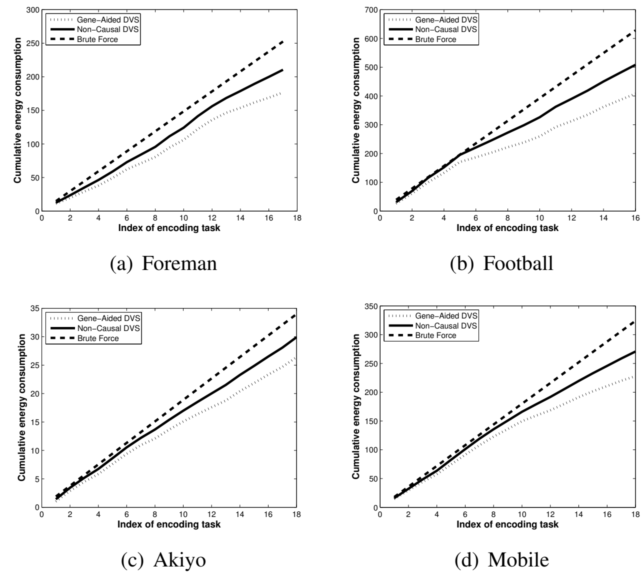
Finally, let us investigate the impact of variance on the optimal scheduling policy. In Figure 3, three frequency scheduling curves are plotted for different variance σ^2 while the same mean μ. It can be seen that, the shape varies for different variances, and larger variance leads to earlier frequency acceleration.



(a) Foreman $\mu = 3Bc, \sigma = 0.23Bc$ (b) Football $\mu = 4Bc, \sigma = 0.41Bc$

(c) Akiyo $\mu = 1.5Bc, \sigma = 0.08Bc$ (d) Mobile $\mu = 3.1Bc, \sigma = 0.24Bc$

Fig. 4. Per-GOP CPU cycle consumption histogram and normal distribution CDF fitting for four different videos



(a) Foreman (b) Football

(c) Akiyo (d) Mobile

Fig. 5. Cumulative energy consumption for three alterative DVS policies

In summary, owing to these characteristics, our optimal DVS solution can be implemented with a low complexity. The frequency expression in Eq.(13) is a light weight computation process, since it can tabulated. In addition, the curve shifting relationship between different workload can reduce the complexity. Therefore, our proposed algorithm is suitable for mobile devices with a limited energy budget.

IV. DVS Application to Real-Time Video Encoding

In this section, we apply our proposed optimal DVS scheduling algorithm to real-time video encoding. Firstly, we evaluate the workload characteristics of H.264/AVC video encoding, and then apply the optimal DVS scheduling policy for it and compare the energy consumption with different distribution estimation configurations.

A. Empirical Workload Distribution for H.264/AVC Coding

In a real-time video encoding system, the real-time management unit can be selected as a single frame or a batch of frames. Considering the high complexity of encoding management, we set a batch of frames encoding process (i.e. encoding of a GOP) to be an individual task as the minimum real-time encoding unit. Several CIF size sample videos are used as test sequences and encoded with x264 software [14], which is a high performance open source H.264/AVC encoder. The raw video sequences are compressed with frame structure of I-B-P-B-P-B...(GOP-16), a frame rate of 25 fps and quantization parameter of 30. The encoding experiments run on a 3GHz Intel Core Duo CPU. The number of cycles consumed is collected via Oprofile tools [15]. The x264 platform-specific assembly optimization is disabled for platform-independent results.

In Figure 4, we plot the histograms of the per-GOP CPU cycles consumed for encoding four different video sources. The resulted histograms are curve-fitted with a (truncated) normal distribution CDF. It can be seen that video encoding workload can be well modeled with a normal distribution.

TABLE I
Energy Saving Over Brute-Force Algorithm

Algorithm	Foreman	Football	Akiyo	Mobile
Gene-Aided DVS	29.7%	35.2%	22.2%	29.4%
Non-Causal DVS	16.4%	18.9%	11.8%	16.2%

B. Optimal Clock Frequency Configuration and Energy Consumption

In this subsection, we first compare the energy performance of the following three scheduling algorithms.

1) Brute-Force Scheduling Algorithm: in this case, the scheduler has access to the overall distribution of all the GOP-encoding tasks (a non-causal estimator) and applies a flat frequency-scheduling policy to meet the encoding delay deadline;
2) Gene-Aided DVS Algorithm: in this case, the scheduler knows the exact number of CPU cycles consumed to encode each GOP and applies a flat-frequency scheduling policy to meet the encoding delay deadline;
3) Non-Causal DVS Algorithm: in this case, the scheduler has access to the overall workload distribution for all the GOP-encoding tasks (a non-causal estimator) and applies our proposed DVS policy to meet the encoding delay deadline with a target task completion probability.

In our first experiment, the task completion probability is set to be 95%. For both the brute force algorithm and the Non-Causal DVS algorithm, the workload distribution is obtained from global experiment data. In Figure 5, we present the simulation results of cumulative energy consumption for the three scheduling algorithms, as a function of the number of encoded GOPs. We first notice that our proposed non-causal

TABLE II
ENERGY CONSUMPTION GAP COMPARED TO GENE-AIDED DVS

Video	Non-causal	Recent-k				
		k=3	k=6	k=9	k=12	k=15
Foreman	18.9%	15.9%	17.7%	16.9%	16.7%	17.1%
Football	25.1%	14.8%	22.4%	24.5%	25.9%	27.2%
Akiyo	13.3%	11.3%	14.2%	14.5%	15.1%	15.4%
Mobile	18.7%	10%	13.9%	16.8%	18.5%	18.5%

DVS algorithm achieves a significant gain compared to the brute Force scheduling algorithm. However, our proposed non-causal DVS algorithm experience some performance penalty from the Genie-Aided DVS algorithm. The average energy saving gain is illustrated in Table I. We can see that the non-causal DVS algorithm can achieve an energy saving gain of $10\% - 20\%$, and the Gene-Aided scheduling algorithm can provide a gain of $20\% - 35\%$.

Our second experiment addresses the issue of workload estimation for DVS. In reality, the non-causal estimator for the workload distribution is infeasible. A more practical approach is to adopt a causal estimator for the workload distribution. In [11], a few sampling methods were investigated for such a purpose. In this paper, we adopt a *Recent-k method* to estimate the empirical workload distribution and evaluate its impact on the energy consumption for real-time video encoding on mobile devices. Specifically, the *Recent-k method* uses the sample of the k most recent encoding tasks to estimate the workload distribution (i.e., the mean and the variance of an embedded Gaussian distribution). The energy consumption penalty ($\delta = (E_{DVS} - E_{Gene})/E_{Gene}$) against the Gene-Aided DVS algorithm is used for performance evaluation. We calculate the energy consumption penalty for the *Recent-k* method and the non-causal DVS algorithm in Table II for $k = 3, 6, 9, 12, 15$. It can be observed that the estimator using the shortest history ($k = 3$ scheme) can achieve more energy saving compared to long history estimation (Global estimation), and thus is more suitable for video encoding. It can be understood as follows. In real-time video encoding application, due to the dynamics of video motion, current frame will only take some of neighboring frames as reference, which leads to the strong correlation on neighboring frames encoding. Therefore, it is reasonable that estimation based on a shorter history can conserve more energy, especially for the fast motion videos, such as "Football".

V. CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of minimizing energy consumption for real-time video encoding on mobile devices via the DVS technology. The problem is formulated as a constrained optimization problem. Under a probabilistic workload model, we obtain closed-form solutions for both the optimal clock frequency configuration and the resulted minimum energy for GOP encoding. Numerical results indicate that our derived optimal solution outperforms the brute-force approach significantly. Moreover, we apply the optimal solution for real-time H.264/AVC video encoding application. Our numerical results suggests that an energy saving of $10\% - 20\%$ can be achieved, compared to the flat clock frequency scheduling.

In this paper, the analytical result for the impact of estimation error is not proposed. Therefore, in the future work, we will do more effort to analyze the tight bound of energy consumption difference while considering the estimation error of workload distribution.

REFERENCES

[1] [Online]. Available: http://www.abiresearch.com/home.jsp

[2] J. Vass, S. Zhuang, and X. Zhuang, "Scalable, error-resilient, and high-performance video communications in mobile wireless environments," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 7, pp. 833–847, 2001.

[3] "Visual networking index: Global mobile data traffic forecast update, 2010-2015," White paper, Cisco Systems, Inc., Feburary 2011, available online.

[4] W. Yuan, K. Nahrstedt, S. Adve, D. Jones, and R. Kravets, "GRACE-1: Cross-layer adaptation for multimedia quality and battery energy," *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, pp. 799–815, 2006.

[5] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14 –23, Oct 2009.

[6] W. Yuan and K. Nahrstedt, "Energy-efficient soft real-time CPU scheduling for mobile multimedia systems," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 149–163, 2003.

[7] Z. He, W. Cheng, and X. Chen, "Energy minimization of portable video communication devices based on power-rate-distortion optimization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 5, pp. 596–608, 2008.

[8] L. Su, Y. Lu, F. Wu, S. Li, and W. Gao, "Complexity-Constrained H.264 Video Encoding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 4, pp. 477 –490, april 2009.

[9] T. Burd and R. Brodersen, "Processor design for portable systems," *The Journal of VLSI Signal Processing*, vol. 13, no. 2, pp. 203–221, 1996.

[10] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, july 2001, pp. 251–259.

[11] J. Lorch and A. Smith, "Improving dynamic voltage scaling algorithms with PACE," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 29, no. 1. ACM, 2001, pp. 50–61.

[12] ——, "PACE: A new approach to dynamic voltage scaling," *Computers, IEEE Transactions on*, vol. 53, no. 7, pp. 856–869, 2004.

[13] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[14] VideoLAN Organization, "x264." [Online]. Available: http://www.videolan.org/developers/x264.html

[15] "Oprofile 0.9.7." [Online]. Available: http://oprofile.sourceforge.net