Hybrid Dynamic Thermal Management Based on Statistical Characteristics of Multimedia Applications

Inchoon Yeo and Eun Jung Kim
Department of Computer Science
Texas A&M University
College Station, TX 77840
{ryanyeo, ejkim}@cs.tamu.edu

ABSTRACT

Recently multimedia applications become one of the most popular applications in mobile devices such as wireless phones, PDAs, and laptops. However, typical mobile systems are not equipped with cooling components, which eventually causes critical thermal deficiencies. Although many low-power and low-temperature multimedia playback techniques have been proposed, they failed to provide QoS (Quality of Service) while controlling temperature due to the lack of proper understanding of multimedia applications. We propose Hybrid Dynamic Thermal Management (HDTM) which exploits thermal characteristics of both multimedia applications and systems. Specifically, we model application characteristics as the probability distribution of the number of cycles required to decode a frame. We also improve existing system thermal models by considering the effect of workload. This scheme finds an optimal clock frequency in order to prevent overheating with minimal performance degradation at runtime.

The proposed scheme is implemented on Linux in a Pentium-M processor which provides variable clock frequencies. In order to evaluate the performance of the proposed scheme, we exploit three major codecs, namely MPEG-4, H.264/AVC and H.264/AVC streaming. Our results show that HDTM lowers the overall temperature by 15°C and the peak temperature by 20°C , while maintaining frame drop ratio under 0.2% compared to previous thermal management schemes such as feedback control DTM [8], Frame-based DTM [5] and GOP-based DTM [15].

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Reliability, Availability, and Serviceability

General Terms

Temperature, Multimedia, DVFS, DTM

ISLPED'08, August 11–13, 2008, Bangalore, India..

Keywords

Dynamic Thermal Management, Thermal Model, Multimedia applications

1. INTRODUCTION

Nowadays, the demand for portable devices with multimedia capability derives embedded systems market. Although modern microprocessors can meet the computation requirement for multimedia data, high power density converts power dissipation into a huge amount of heat as a feature size decreases. According to [4], this rising system temperature affects the system reliability by doubling the system failure rate when the temperature increases by 10°C. Therefore, it is critical to keep the temperature of the microprocessor under safe limits during runtime.

Dynamic Voltage/Frequency Scaling (DVFS) is a common method to control temperature in microprocessors. However, due to the nature of multimedia applications with different frame sizes and types in data, it is not easy to match their QoS requirements while temperature is under control. There have been handful studies on temperature management for multimedia applications [12, 10, 7, 8, 13, 15]. However, we found out that these schemes tend to overestimate or underestimate multimedia application requirements, which could result in false, inevitably leading to high operation temperature or performance degradation.

In order to compensate the tradeoff between performance and temperature control, in this study, we first derive application characteristics in various multimedia applications by transmitting MPEG-4, H.264/AVC, and H.264/AVC stream over networks. Application characteristics can be represented by *cycle demand*, which is the number of cycles required to decode a frame. Using this representation, we estimate application characteristics of input multimedia data more accurately based on the probability of cycle demand. Then, we propose Hybrid Dynamic Thermal Management (HDTM) that takes optimal frequencies to avoid thermal emergency while minimizing performance degradation. In order to find the optimal frequency, HDTM estimates operational temperature in both proactive and reactive steps. In the proactive step, we enhance existing system thermal models by considering the effect of workload. After profiling for workload, we obtain thermal parameters in a specific processor. In the reactive step, we obtain the probability distribution of cycle demand at runtime.

We experimented on an Intel's Pentium-M processor using 27 multimedia data. Compared to feedback control

DTM [8], Frame-based DTM [5] and GOP-based DTM [15], HDTM lowers average temperature by 15°C and peak temperature by 20°C or more, with maximum 0.2% frame drop ratio. The main contributions of this paper are summarized as follows:

- We estimate multimedia application's thermal characteristics with only 2.5% error on average.
- Compared to the previous DTMs such as feedback control [8], Frame-based [5], and GOP-based [15], our proposed HDTM lowers temperature by 15°C on average when running MPEG-4, H.264/AVC, and even streaming H.264/AVC under 0.2% frame drop ratio.
- We provide a hybrid estimated scheme with a reactive method using statistical cycle demand information and a proactive method for system temperature behavior with a certain workload.

The remainder of the paper is organized as follows : The existing DTM is introduced in Section 2, and the design and algorithm for our proposed HDTM in multimedia applications are explained in Section 3. In Section 4, the implementation and analysis results are discussed and conclusions are provided in Section 5.

2. RELATED WORK

The dynamic thermal managements in multimedia systems can be classified into proactive methods [13, 10, 7] and reactive methods [12, 8, 15, 5]. In proactive methods, J. Pouwelse *et al.* estimated the decoding time per frames based on the offline information, such as decoding time and frame size [10]. In [7], Nurvitadhi *et al.* compared the performance of three different Dynamic Voltage Frequency Scaling (DVFS) schemes in multimedia models including per-GOP, Direct and Dynamic where decoding time of each frame was calculated based on previous frames of same type. Although their approach estimated the decoding time for incoming frame based on information of decoding macro-blocks, their method did not provide a solution for avoiding thermal emergency. In [13], Srinivasan *et al.* proposed the predictive thermal management using the profiled information. Their method showed maximum performance under thermal constraints.

In contrast to proactive methods, reactive methods determine future frequency of the system based on the historical information. The results of previous task configure future frequency for next frames in multimedia system. Feedback control was proposed for multimedia systems [8]. The control module changed the level of frequency based on the display buffer occupancy. However, since the method is designed only for power control, there is no consideration of temperature. Frame-based DTM takes advantage of the decoding time of the current frame to adjust more specific frequency [5]. This method can be vulnerable to overestimation or underestimation of multimedia application requirement when the decoding time changes rapidly [5]. Also, since this method depends on the current decoding time, it is unable to predict future temperature or thermal characteristics. In [12], a predictor estimates the decoding time of an incoming Group of Pictures (GOP) based on the relationship between frame size and decoding time of past frames. The calculated decoding time determines future frequency and voltage for the future GOP. However, their method has been only exploited in the context of power control. In order to control temperature as well as power, Yeo *et al.* proposed that future frequency be decided for next GOP under temperature constraints according to the complexity of the current GOP in [15].

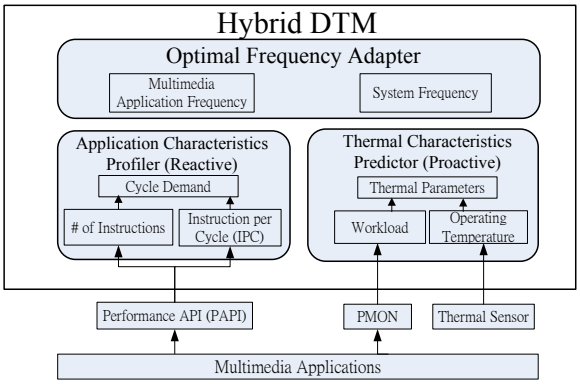


Figure 1: HDTM overview

In summary, proactive methods achieve energy saving and more accurate temperature management, but require the pre-processing data and overhead, while reactive methods control temperature using historical information, but are triggered when processors are already overheated.

3. HYBRID DYNAMIC THERMAL MANAGEMENT (HDTM)

In this section, we present our hybrid method, Hybrid Dynamic Thermal Management (HDTM), to integrate both proactive and reactive methods. With the proactive method, our proposed HDTM estimates system thermal characteristics according to workload before running multimedia applications. Since system thermal characteristics by using thermal parameters is dependent on a specific processor or architecture, thermal characteristics can be measured by thermal model added the effect of workload. With the reactive method, we obtain the probability distribution of cycle demand at runtime, which is the number of cycle required to decode a frame. Those proactive and reactive information are used to determine an optimal frequency in multimedia applications. As shown in Figure 1, HDTM consists of three components: an application characteristics profiler as the reactive method, thermal characteristics predictor as the proactive method, and optimal frequency adaptor. We describe the operations of each component in the following sections.

3.1 Application Characteristics Profiler

The application characteristics profiler estimates the probability distribution of cycle demand for decoding frames at runtime. We estimate the cycle demand distribution to obtain more accurate multimedia computation requirements.

To estimate the cycle demand distribution of decoding frames at runtime, we need two steps: the first step is to measure cycle usage measured by Instruction Per Cycle (IPC) and the number of instructions in a fixed window size, and the second step is to derive the probability distribution at runtime. In order to measure cycle usage for

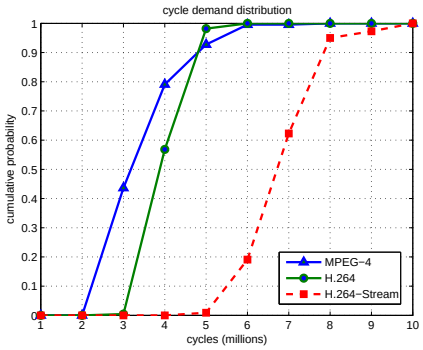


Figure 2: The cumulative distribution function (cdf) of decoding frames in the multimedia application.

decoding frames at runtime, we use Performance Monitoring Unit (PMU) for a Pentium-M and add a Instruction Counter and IPC measurement module into a decoding part of multimedia applications [9]. As a decoding step executes, executed cycles are calculated by Equation (1).

$$C_i = \frac{I_i}{IPC_i}, \tag{1}$$

where C_i is the used cycles, I_i is the number of instruction for decoding a frame, and IPC_i is the value of IPC for decoding i_{th} frame obtained by PMU. Next, we can derive the probability distribution of cycle demands in a fixed window size, which is equal to frames per second (fps). Let C_{min} and C_{max} be the minimum and maximum numbers of cycles, respectively, in the window. In our environments, C_{min} and C_{max} are assumed to be 1 million cycles and 10 million cycles because the most multimedia applications requires meeting 96% of frame decoding demands no more than 9 million cycles, and then 9 million cycles per frame is the maximum requirement for decoding in multimedia applications [16]. We obtain a probability density function (pdf) and a cumulative distribution function (cdf) using following steps:

1. We denote the cdf as $F(x)$ for a random variable X as the number of cycles for decoding a frame according to a pdf $f(x)$ and probability p using Equation (2)

$$P(C_{min} \leq X \leq C_{max}) = \int_{C_{min}}^{C_{max}} F(x)dx, \tag{2}$$

   where X in the interval $[C_{min}, C_{max}]$ with the same sized groups and $F(x) = P(X \leq x) = \sum_{y:y \leq x} p(y)$. We refer to $c_0, c_1, \cdots, c_n$ with the same size, 1 million, as the group boundaries.

2. For decoding frames in multimedia application, we estimate the probability $P(C_{min} \leq X \leq C_{max})$ in MPEG-4, H.264/AVC, and H.264/AVC streaming, as shown in Figure 2.

To satisfy a various computational requirements, a frequency for decoding frames should be decided by cycle demand based on the probability requirements for decoding frames in the window. Specifically, let ρ be the probability required for decoding frames in a window, and every decoding task for a frame needs to meet the probability ρ of deadlines. In other words, every frame of the window should meet its deadline with a probability ρ. To support this requirement, the C_k cycles should be allocated to all decoding tasks in the same window, i.e.,

$$F(C) = P[X \leq C_k] \geq \rho. \tag{3}$$

To determine this parameter C_ρ for a task, we find C_x whose cumulative distribution is at least ρ, i.e., $F(C_x) = P[X \leq C_x] \geq \rho$. Since we assume the probability ρ is 0.96, we determine this C_x as the parameter C_ρ. In order to get a frequency for decoding frames at a given window, frequency, f_d, can be obtained by Equation (4).

$$f_d = \frac{C_\rho \times fps}{\Delta t}, \tag{4}$$

where f_d is a frequency for cycle demand for decoding frames in the window, fps is frames per second, and the time interval Δt is 1 sec. As in Figure 3(a), the demanded number of instructions shows the requirement of instructions are different according to frames. We determine an optimal frequency by the number of instructions for decoding frames using a real multimedia data as shown in Figure 3(b).

Even though we find an optimal frequency for decoding frames as shown in Figure 3(b), additional system workload generated by the operating system such as scheduling overhead, file I/O handling, and network monitoring should be considered to guarantee the performance in real systems. The system workload occupies between 5% and 50% according to assigned frequency. Therefore, the optimal frequency (f_d) for decoding frames should be adjusted by including system workload. We present this issue in Section 3.3.

3.2 Thermal Characteristics Predictor

In this paper, we only consider dynamic power which is a dominant factor. The static power or leakage power can be ignored in temperature control [2].

3.2.1 Thermal Model

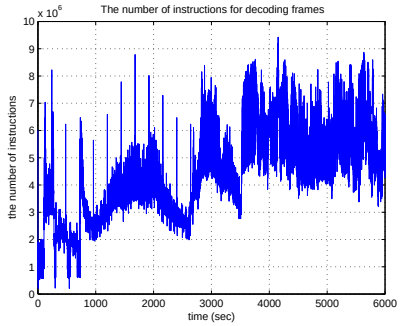
We consider a thermal model of a processor in the relationship between processor frequency and temperature [2, 6]. By modeling the power dissipation, more precise models can be derived from a simple model [14]. We analyze Fourier's Law of heat conduction where the formula states that the rate of heating or cooling is proportional to the difference in temperature between the object and the environment [14]. We define $T(t)$ and $P(t)$ as temperature and power at time t, respectively. Then we can use the Fourier's Law as the following [3, 6]:

$$T'(t) = P(t) - bT(t), \tag{5}$$

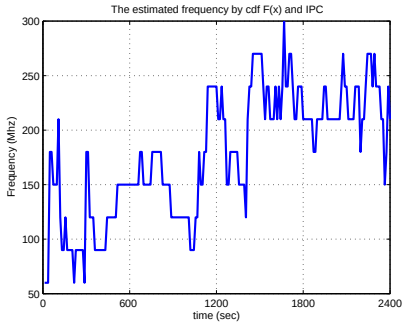
where b is a positive constant representing the power dissipation rate. Now, we define $f(t)$ as processor frequency at time t. Since the power consumption of a processor is an increasing convex function of the frequency, power consumption can be represented by frequency [3]. Most studies assume that power and processor frequency are relevant to the followings:

$$P(t) = a(f^\alpha(t)), \tag{6}$$

for some constant a and $\alpha > 1$. With an assumption that $T_0 = 0$ (the initial temperature is the ambient one), the solution of Equation (5) using Equation (6) can be presented as follows:



(a) The number of instructions



(b) The estimated frequency

Figure 3: The number of instruction and the estimated frequency by cdf $F(x)$ and IPC without considering system workload

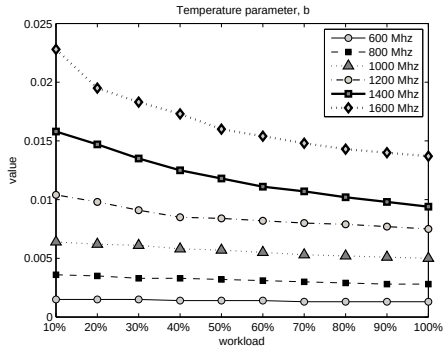


Figure 4: The value of temperature parameter (b) by frequency and workload

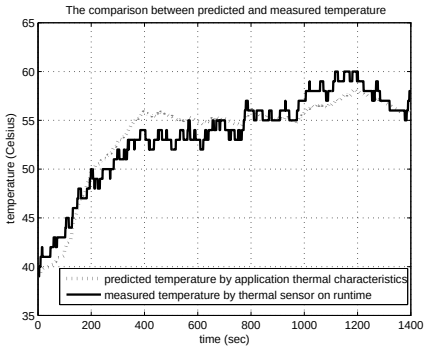


Figure 5: The comparison between predicted and measured temperature in multimedia application

$$T(t) = \int_{t_0}^{t} a(fr^{\alpha}(\tau)e^{-b(t-\tau)})d\tau + T_0 e^{-b(t-t_0)}. \quad (7)$$

We can derive the following equation if we maintain the frequency constant at $f(t) = f_c$ during the time interval at $[t_0, t]$.

$$T(t) = \frac{a(f_c^{\alpha})}{b} + (T(t_0) - \frac{a(f_c^{\alpha})}{b})e^{-b(t-t_0)}, \quad (8)$$

where f_c is the current frequency on the processor. In order to determine thermal parameters, a and b, we assume $\alpha = 3.0$ [3], and then we can obtain the values for a and b.

3.2.2 System Thermal Characteristics

In this section, we illustrate the relationship between the change in temperature and thermal parameters a and b in detail. The change in temperature is based on individual component's thermal resistance and capacitance in specific processors [11]. To obtain current and future temperatures, we should take account for thermal resistance R_{th} and thermal capacitance C_{th}, while changing in temperature from T_{old} to T_{new} over a time interval Δt like Equation (9).

$$T_{new} = P \cdot R_{th} + (T_{old} - P \cdot R_{th})e^{\frac{-\Delta t}{R_{th} \cdot C_{th}}}. \quad (9)$$

With Equation (9) and (8), we derive the thermal parameters a and b as follows:

$$a = \frac{1}{C_{th}}, \; b = \frac{1}{R_{th} \cdot C_{th}} \quad (10)$$

By Equation (10), thermal parameter a is represented as thermal capacitance C_{th}. Thermal capacitance is defined as the amount of thermal energy required to raise temperature of one mole of material by 1 Kelvin and can be measured at constant volume or at constant pressure [6]. Therefore, this value is practically constant in the same material. In contrast with the value a, the other thermal parameter b is related to application's workload. This is because thermal resistance is in inverse proportional to power consumption. As shown in Figure 4, thermal parameter b has much to do with application's workload as well as the value of thermal resistance and thermal capacitance.

With thermal parameters, we estimate future temperature for multimedia applications at runtime, and the accuracy of our thermal characteristics predictor has 2.5% error in average compared to measured temperature by thermal sensor. Since future temperature is estimated by thermal characteristics according to dynamic workload, the application thermal characteristics can be tracked as depicted in Figure 5.

3.3 Optimal Frequency Adaptor

In this section, we introduce how to find an optimal frequency using both application workload and system workload. As mentioned in Section 3.1, we obtain an optimal frequency based on an application demanded frequency (f_d) for the probability of demanded cycles in a multimedia application. However, it cannot guarantee the performance in

Table 1: The movie data for experiments

Movie Title	fps	Scene complexity
Star Wars 3	23	*High*
Terminator 3	23	*High*
Any Given Sunday	23	*High*
24 (Season 1)	23	*High*
Under World 2	25	*High*
Blue Storm	23	*Mid*
Transporter 2	25	*Mid*
Eragon	25	*Mid*
Gilmore Girls	23	*Low*

multimedia applications because a system workload generated by operating system is neglected. The system workload, W_s, can be calculated by Equation (11).

$$W_s = W_t - W_d, \tag{11}$$

where W_t is the current total workload, W_s and W_d are the system workload and the application-demanded workload, respectively. Also, f_c is the current frequency for a processor in multimedia applications (i.e., the range of frequency is between 600 Mhz and 1600 Mhz in this study). Therefore, the system required frequency, f_s, can be calculated by Equation (12).

$$f_s = \frac{W_t - W_d}{W_t} \times f_c. \tag{12}$$

Therefore, the optimal frequency, f_{opt}, can be derived as follows:

$$f_{opt} = f_d + f_s, \tag{13}$$

where f_d is an application required frequency and f_s is the system required frequency, respectively. With the optimal frequency f_{opt}, future temperature can be estimated by thermal characteristics. If future temperature reaches a thermal threshold (defined as 70 °C in this study), the optimal frequency, f_{opt}, should be readjusted. However, our results always maintain temperature under the thermal threshold.

4. EXPERIMENTAL RESULTS

For our experiments, we modified a multimedia player source code and measured temperature using ACPI on Linux. All experiments were performed without any cooling components, and the range of frequency had six levels from 600 Mhz to 1600 Mhz. We used 27 multimedia data encoded by MPEG-4, H.264/AVC, and streaming H.264/AVC over network. And we used Darwin Streaming Server for H.264/AVC streaming service [1]. These movies were chosen as representatives of three types of complexity; high-complexity, mid-complexity, and low-complexity. These movies were presented for 30 minutes which had about 40,000 frames encoded without any alteration of size and *fps* of the original data into MPEG-4, H.264/AVC and H.264/AVC streaming formats. Table 1 summarizes all movie data used in this study.

Also, we measured the number of instructions and IPC using the Performance API (PAPI) based on performance counter in most major microprocessors [9]. We compared temperature levels of the HDTM with three previous DTMs. The feedback control DTM controls frequency based on feedback control for frame buffers. According to the buffer occupancy, the frequency may increase to accommodate required decoding performance, or decrease for energy saving purposes. In this case, this system cannot always guarantee the immediate necessary frequency. For example, if there is sufficient buffer occupancy, the frequency maintains to be low and in turn, the decoding time increases. Due to the elongated decoding time, the buffer occupancy will decrease or increase the chances of dropping frames. This weakens the feedback control scheme from providing an optimal frequency immediately. Therefore, the feedback DTM maintains relatively higher temperature levels compared to other DTM schemes, as depicted in Figure 6.

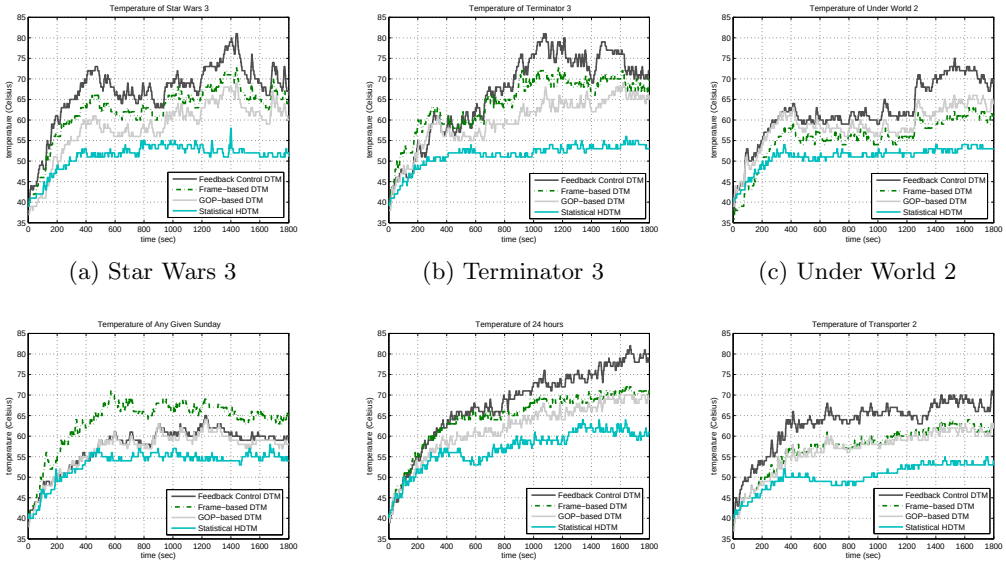
Frame-based DTM controls frequency based on the current decoding time of each frame. Frame-based DTM takes advantage of the decoding time of the current frame to adjust the frequency. This scheme can be vulnerable to multimedia applications where the decoding time changes rapidly and cause to increase overhead in the overall system. Also, since this scheme depends on the current decoding time, it is unable to predict future temperatures or thermal characteristics.

GOP-based DTM controls frequency based on the information of several frames that consist current GOP. The GOP-based DTM overcomes the disadvantages of feedback control and frame-based DTM by using the GOP information in order to adjust the frequency accordingly. Specifically, this scheme prevents any delays in applying different frequencies as in the feedback control DTM and prevent potential overheads from frequent changes in the frequency unlike the frame-based DTM. This is because the GOP-based DTM has superior performance compared to the feedback controlled DTM and the frame-based DTM but maintains higher temperature levels than the proposed HDTM. The reason is hard to estimate future temperatures or to determine the optimal frequency due to insufficient information from GOP.

The proposed HDTM derives statistical information by taking advantage of the cycle demand obtained by IPC and the number of instructions. Based on the statistical information for the previous frames, this scheme calculates the currently required frequency.Also, this leads HDTM to operate in lower temperature levels than other DTM schemes as shown in Figure 6. HDTM based on application thermal characteristics lowers temperature by about 15°C in average and reduces up to 20°C in peak temperature compared to other previous DTMs. Since HDTM meets up to 0.2% frame drop ratio in multimedia applications, HDTM outperforms the previous DTMs, the feedback control DTM, the Frame-based DTM, and the GOP-based DTM.

5. CONCLUSIONS

In this paper, we propose Hybrid Dynamic Thermal Management (HDTM) which uses both application characteristics represented by the probability distribution of cycle demand to decode a frame and system thermal model augmented by the effect of workload. Our experimental results show that the distribution of cycle demands in various codecs affect temperature directly as application workload. This implies that the overall temperature can be predicted and controlled by the optimal frequency to decode frames for any type of multimedia data. Also, the proposed HDTM



(a) Star Wars 3 (b) Terminator 3 (c) Under World 2

(d) Any Given Sunday (e) 24 hours (f) Transporter 2

Figure 6: Resulting temperatures with Feedback control DTM, Frame-based DTM, GOP-based DTM, and Statistical HDTM

scheme explores application thermal characteristics based on statistical information of cycle demands that can estimate future temperature within 2.5% prediction error in average compared to the measured temperature by a thermal sensor. Therefore, HDTM provides more accurate estimation and more efficient temperature management compared to other schemes such as feedback control DTM, Frame-based DTM, and GOP-based DTM.

6. REFERENCES

[1] Apple, "Darwin Streaming Server," available from http://developer.apple.com/.

[2] N. Bansal, T. Kimbrel, and K. Pruhs, "Speed Scaling to Manage Energy and Temperature," *Journal of ACM*, vol. 54, no. 1, 2007.

[3] N. Bansal, T.Kimbrel, and K. Pruhs, "Dynamic Speed Scaling to Manage Energy and Temperature," in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[4] T. Brady, D. Bodas, D. Gabel, B. Griffith, T. Niemela, and D. Perchlik, "PC Power Consumption A Challenege and Opportunity," in *International Symposium on Electronics and the Environment (ISEE)*, 2000.

[5] K. Choi, K. Dantu, W. C. Cheng, and M. Pedram, "Frame-based Dynamic Voltage and Frequency Scaling for a MPEG Decoder," in *International Conference on Computer Aided Design (ICCD)*, 2002.

[6] J.E.Sergent and A.Krum, *Thermal Management Handbook*. McGraw-Hill, 1998.

[7] B. Lee, E. Nurvitadhi, R. Dixit, C. Yu, and M. Kim, "Dynamic Voltage Scaling Techniques for Power Efficient Video Decoding," *the EUROMICRO Journal*, vol. 51, no. 10-11, pp. 633–652, 2005.

[8] Z. Lu, J. Lach, M. Stan, and K. Skadron, "Reducing Multimedia Decode Power using Feedback Control," in *International Conference on Computer Aided Design (ICCD)*, 2003.

[9] PAPI, "Performance API," available from http://icl.cs.utk.edu/papi.

[10] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor," in *MobiCom*, 2001.

[11] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-Theoretic Techniques and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management," in *International Symposium on High-Performance Computer Architecture (HPCA-8)*, 2002.

[12] D. Son, C. Yu, and H. N. Kim, "Dynamic Voltage Scaling on MPEG Decoding," in *International Conference on Parallel and Distributed Systems (ICPADS)*, 2001.

[13] J. Srinivasan and S. Adve, "Predictive Dynamic Thermal Management for Multimedia Applications," in *International Conference on Supercomputing (ICS)*, 2003.

[14] S. Wang and R. Bettati, "Reactive Speed Control in Temperature-Constrained Real-Time Systems," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2006.

[15] I. Yeo, H. K. Lee, K. H. Yum, and E. J. Kim, "Effective Dynamic Thermal Management for MPEG-4 Decoding," in *International Conference on Computer Aided Design (ICCD)*, 2007.

[16] W. Yuan and K. Nahrstedt, "Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems," in *ACM Symposium on Operating Systems Principles (SOSP)*, 2003.