PROCESSOR LOAD ANALYSIS FOR MOBILE MULTIMEDIA STREAMING: THE IMPLICATION OF POWER REDUCTION

Min LI[1,2,3], Xiaobo WU[1], Zihua GUO[2], Richard YAO[2], Xiaolang YAN[1,3]
[1]*Institute of VLSI Design, Zhejiang Univ., P.R.China*
[2]*Microsoft Research Asia*
[3] *School of Information Science and Engineering, Zhejiang Univ., P.R. China*
limin@imec.be

ABSTRACT

The software codec on mobile device introduces significant power consumption because the energy efficiency of general processor based system is much lower than that of the dedicated hardware such as ASIC based accelerator. Dynamical Voltage Scaling (DVS) is one of the most efficient techniques to promote the energy efficiency. Most existing papers on this topic use simple heuristics to predict processor load, and poor prediction accuracy is observed in experiments. We advocate intensive analysis on processor load before designing DVS framework and algorithm. Hence, we conduct load analysis on more than 600 processor load trace files for 57 test sequences and 98 representative clips from Internet. Basic statistical analysis and time series analysis are applied intensively to identify major characteristics of the processor load. The analysis shows that it is feasible to predict processor load using low order linear time series model if the load is sampled using feature period. Moreover, there is indeed significant potential to reduce the energy consumption. Based on the analysis results, we develop a fully adaptive DVS technique to adjust supply voltage online with controllable penalty.

1. INTRODUCTION

Multimedia streaming has been shown as one of the most popular applications for future mobile device, especially in Asian market. In conjunction with 3G and Wi-Fi, multimedia streaming will be widely deployed in the near future.

Although some semiconductor vendors have fabricated handset chipsets with integrated codec hardware unit, software based codec is also preferred because of its flexibility and reconfigurability. Many recently released mobile processors (such as Intel PXA series and Cirrus Logic EP9312) provide enough processing power for software codec.

The streaming client has several metrics, such as media quality, UI, energy consumption, etc.. The energy consumption has become the most critical one on mobile devices. The energy consumption of software codec largely comes from intensive memory access and processor activity. On the memory side, there is a lot of existing work focusing on optimize access pattern and promote data reuse [1]. On the processor side, Dynamical Voltage Scaling (DVS) has been studied to low down the supply voltage of processor when it is not fully loaded [2][3][4]. Actually DVS also has significant impact on the energy efficiency of memory system, because the clocking of internal memory, like Cache and Scratch Pad Memory, is usually coupled with processor core. Most existing papers are based on simple heuristics, and none of them perform comprehensive analysis on the processor load. Poor performance is observed in experiments. We advocate performing intensive and comprehensive analysis on processor load, so that energy efficiency optimization can be built on solid foundation instead of an Ad Hoc one.

The contributions of our work are as following: (1) We comprehensively analyze the processor load for mobile multimedia streaming. Intensive statistical analysis is applied to identify major characteristics of the processor load. The analysis result shows that low order linear time series based processor load prediction is feasible if the load is sampled using *feature period*, and there is indeed significant potential to reduce energy consumption. (2) Based on the analysis result, we develop a fully adaptive DVS framework and a set of algorithms to adjust supply voltage online. Accurate prediction is achieved with standard error of deviation below 7%, and more than 50% energy reduction is achieved with penalty below 10% when streaming CIF sequences.

The rest of the paper is organized as five sections. Section 2 gives the detail of experiments platform and collected traces. Section 3 presents the result of basic statistical analysis. Section 4 analyzes the feasibility of time series based processor load prediction. Section 5 briefs the experiments of load prediction and our DVS framework. Finally, section 6 gives the conclusion.

2. THE EXPERIMENT AND FEATURE PERIOD

In our experiments, the hardware platform is an ESM-2615 System on Module (SOM) [5]. We equip the board

0-7803-9332-5/05/$20.00 ©2005 IEEE

with 32M SDRAM, 64M solid-state storage card and a Lucent WaveLan wireless LAN card. Actually this type of SOMs have been widely used in web pad, wireless terminal, industrial handheld, HMI and so on.

There are several prevailing mobile operating systems in the market: Microsoft Windows Mobile Family, Symbian, Palm and Linux. We use Windows CE 4.2 in our experiments. In fact Microsoft Pocket PC 2003 and Smartphone 2003 have the same kernel features as Windows CE 4.2. When building binary image for the target, we choose a predefined configuration, which enables MPEG4, MP3, WMV/WMA streaming/playback features, and WLAN support.

The processor load x_k is defined as: in the *k-th* T ms interval, percentage of time that the processor spent in active state (running processes/threads). In order to record the traces of processor load, we developed a tracing tool that is tightly coupled with operating system kernel.

First we would like to emphasize one of the most important observations. When transforming traces using FFT, we find that different streaming sources cause different frequency characteristics of processor load, and the characteristics remain relatively stable over certain length of time. It is to say, the processor load has certain *Featured Period (FP)* depending on the media stream. Note that media encoding/decoding is a typical periodical application because the stream is processed in a frame-by-frame manner. At the first glance, the featured period of the processor load seems to be $p = (1000/FPS)ms$. However, it does not always follow this principle. Various optimization techniques were intensively applied in various codecs to reduce memory operation, page pre-charge, cache misses, etc. [1]. Hence, the FP can only be identified by FFT.

Two sets of streaming sources are considered when collecting processor load traces. The first set includes clips from public websites. These streaming sources cover wide range of formats, including different video/audio combination. This set includes 98 streaming sources. The second set consists of well-known video test sequences, such as Container, Foreman, Mobile, News, Paris and Tempete. All test sequences are encoded by Sonic Foundry Vegas using different size, formats and FPS. The set includes 57 streaming sources. For each streaming source, traces are recorded using 15ms, 20ms, 25ms, 30ms and FP. For convenience, we use the string format like Mobile_CIF_V9_25FPS_25ms to name the accordant processor load trace file. In the string, the first part is the name of the test sequence, the second part is the size, the third part is the encoding format, the forth part is the FPS, and the final part is the sample interval with unit being millisecond.

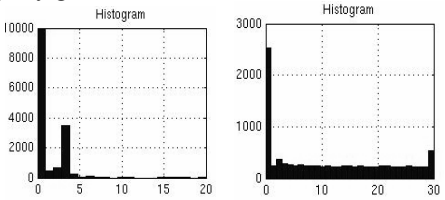
3. BASIC STATISTICAL ANALYSIS

The potential of applying DVS is determined by the average load of processor, and lower load implies higher reachable energy reduction. Almost all traces show surprisingly low mean value, even for a 30FPS WMV9 CIF test sequences, the mean load is around 0.3, which indicates a large room for DVS to save energy. In Table 1 we list the mean for some load traces.

Table 1 Mean of Processor Load

No.	Mame	Size	V.Format	FPS	Mean
1	Container	CIF	MPEG4	20	0.1548
2	Mobile	CIF	MPEG4	30	0.2529
3	News	QCIF	WMV8	25	0.0359
4	Paris	CIF	MPEG4	10	0.1847

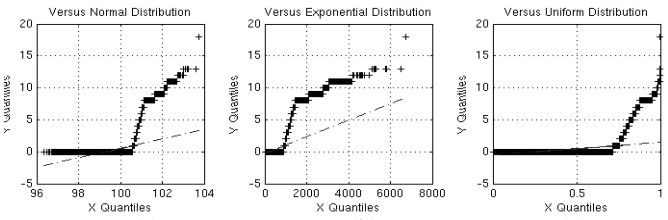
Besides statistics summary, we next try to figure out the Probability Distribution Function (PDF) of the traces. A usual way to identify the PDF is to draw histogram. Besides the usual histogram on the whole data set, we also draw histogram on sliding window on time axis. An example is shown in Fig. 1, where (a) is the histogram of Container_CIF_MPEG_20FPS_20ms, (b) the histogram of Paris_CIF_WMV9_30FPS_20ms. Apparently, there is no evidence in the shape of histogram showing that they following any general distribution.



(a) (b)

Fig. 1 Histogram of Processor Load

Also, we use Quantile-Quantile plot (Q-Q plot) to identify the PDF. The Q-Q plot is a graphical data analysis technique to compare the distributions of 2 data sets. If the two samples do come from the same distribution (same shape), even if one distribution is shifted and re-scaled from the other, the plot will be linear (match the strait line shown in the figure).



(a) (b) (c)

Fig. 2 Q-Q plot. (a) Versus normal distribution; (b) Versus exponential distribution; (c) Versus uniform distribution.

Using Q-Q plot, we try to fit data in trace file to two distributions: exponential and normal distribution, which are widely taken as assumption in some existing work. The Q-Q plots of Mobile_CIF_MPEG_30FPS_25ms is shown in Fig. 2 versus (a) normal distribution, (b) exponential distribution and (c) exponential distribution. Results show that the traces do not follow any of them. This implies that general distribution based techniques are not applicable.

4. TIME SERIES ANALYSIS

The motivation of applying time series analysis is to investigate whether or not the processor load is predictable using time series models. If processor load is predictable, we can adjust the frequency to fully stretch the load onto the whole period, hence achieve the most energy reduction.

The Auto Regressive Moving Average process $ARMA(p,q)$ is defined as

$$x_t = \sum_{i=1}^{p} \varphi_i x_{t-i} + \sum_{i=0}^{p} \theta_i \varepsilon_{t-i}, \theta_0 = 1, \quad (1)$$

where φ_i and θ_i are fixed constants, and ε_t is a sequence of independent random variables with zero mean and variance σ^2. The sequence of ε_t is usually referred as white Gaussian noise $Z = WN(0,\sigma^2)$. The Moving Average process of order q, denoted as $MA(q)$, is a subset of $ARMA(p,q)$ with $p = 0$. The Auto Regressive process of order p, denoted as $AR(p)$, is defined similarly.

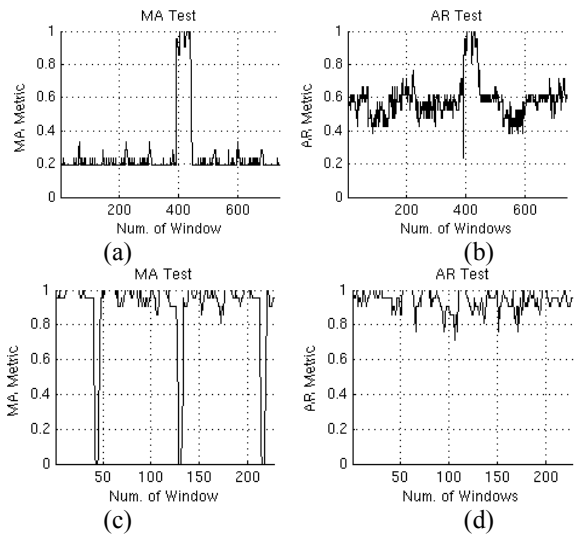


Fig. 3 Visual Inspection for Model Identification. (a)(b): MA and AR metrics when using Ad Hoc sample period; (c)(c): MA and AR metrics when using FP.

When studying a stochastic process, if the original process $Y = \{y_1, y_2, y_3, ..., y_n\}$ is not stationary, we can look at the first order differential process:

$$X = \nabla Y = \{y_2 - y_1, y_3 - y_2, ..., y_{k+1} - y_k, .., y_n - y_{n-1}\}, \quad (2)$$

or the second order differential process: $X = \nabla^2 Y = \{y_{k+2} - 2y_{k+1} + y_k\}$, and so on. If we can find that the differential process is a stationary process, we can look for an $ARMA(p,q)$ model for that.

The process Y is said to be an Auto Regressive Integrated Moving Average process $ARIMA(p,d,q)$, if $X = \nabla^d Y$ is an $ARMA(p,q)$ process.

Since $ARMA(p,q)$ based prediction has big overhead in terms of model fitting (parameters estimation) and real time predicting, high order $ARIMA(p,d,q)$ is useless for online processor load prediction. Hence, we consider only $ARIMA(p,d,q)$ with $p \le 5$ and $q \le 5$.

An ideal $AR(p)$ series has the property that ACF is zero after p lag, but $PACF$ decay exponentially after p lag. On the contrary, an ideal $MA(q)$ series has the property that $PACF$ is zero after q lag, but ACF decay exponentially after q lag. Hence, for a rough identification of linear time series model of series x_t, if the dimension of $\{\hat{\rho}_k \,|\, |\hat{\rho}_k| < 2/\sqrt{N}, q < k \le M\}$ is larger than $0.95 \times (M - q + 1)$, we can assert x_t is a $MA(q)$ series. The same method can be applied to rough identification of the $AR(p)$ process. Based on this principle, we use the following index for identifying time series:

$$u_{MA(q)}(x_t) = \frac{\left|\left\{\rho_k \,|\, |\rho_k| < 2/\sqrt{N}, q < k \le M\right\}\right|}{M - q + 1}, \quad (3)$$

$$u_{AR(p)}(x_t) = \frac{\left|\left\{\varphi_k \,|\, |\varphi_k| < 2/\sqrt{N}, p < k \le M\right\}\right|}{M - p + 1}. \quad (4)$$

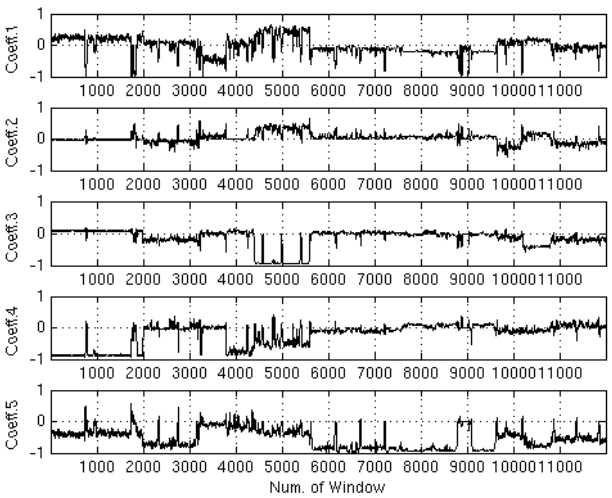


Fig.4. Seasonally Stationary Coefficients of AR Model.

If the index is larger enough, we can assert that the process is MA or AR process.

When applying the time series analysis method to traces recorded at predetermined interval, we found that the traces never show stationary behavior of *MA* or *AR* even after imposing 1-5 times of differential operation. Fig. 3 (a)(b) are the result of original Mobile_CIF_V8_30FPS_25ms.

Then, the same time series analysis method is applied to the traces recorded using our featured period, the results are totally different. Most traces show behavior close to ideal $ARIMA(p,d,q)$ series after the first order differential operation. Fig. 3 (b)(d) are the result of Mobile_CIF_V8_30FPS.

After identifying the model type, the next step is to estimate the parameters of the chosen model (model fitting). There are many methods that are applicable to $ARMA\ (p,q)$ parameters estimation, and the method of moment is adopted in our research. We came to two conclusions after fitting to time series in traces. The first is that all parameters remain relatively stable; the second is that the variance σ^2 of white noise is always very small. Usually it is below 1% when normalized. Hence, the model can be simplified as a low order $AR\ (p)$.

Moreover, we find that the coefficients of the $AR\ (p)$ model are seasonally stationary. As shown in Fig. 4, the coefficients keep almost constant after abrupt change.

5. PROCESSOR LOAD PREDICTION

Based on concrete observation and solid analysis, high prediction accuracy is achieved in experiments. In many previous papers, prediction accuracy is evaluated by mean value of prediction error (PE), However, the standard deviation of PE is much more important, because it represents exactly how much the PE deviates from the mean value. Using our approaches, the PE mean is nearly zero, and PE standard deviation ranges from 2% to 12%. Using AVG [3] based approach, PE mean is also nearly zero, but the standard deviation ranges from 12% to 30%. Using PID [2] control based approach, PE mean is also very low, and standard deviation ranges from 10% to 35%. Typical cases are presented in Table 2.

Table 2. Prediction Accuracy

Source	Our Apporaches		AVG		PID	
	Mean	Std.Err	Mean	Std.Err.	Mean	Std.Err
Tempete,CIF,v9, 30FPS	0.0001	0.0617	0.0165	0.2775	0.0001	0.2559
Paris,CIF,MPEG, 15FPS	0.0001	0.0616	0.0001	0.1262	0.0001	0.1763
Mobile_CIF_v9_30FPS	0.0001	0.0584	0.0001	0.1773	0.0001	0.1895

Since the processor load prediction is quite accurate, prediction based DVS is feasible. We develop a penalty controllable DVS framework and a set of algorithms. In experiment, More than 50% energy reduction is achieved when streaming CIF sequences with deadline miss rate below 10%.

6. CONCLUSION

In this paper we introduce the intensive and comprehensive analysis of processor load of mobile multimedia. The result of statistics analysis shows very low mean processor load, and this implies that DVS is promising for power reduction. This is because processor spends a lot of time in idle state, we can lower down supply voltage and utilize the idle time to reduce energy.

From the result of distribution fitting using Q-Q plot, we can make the conclusion that assumption of regular distribution function (Normal or Exponential) is unrealistic. General probability based method is not applicable.

Time series analysis shows processor load is not a stationary linear time series if the load is sampled using predetermined period, but if the load is sampled using individual featured period, it will be a low order $ARIMA(p,d,q)$. Furthermore, the parameters do not vary frequently. Hence, the time series model based load prediction is feasible. Furthermore, since the variance of the white noise is very small, we can ignore the $MA(p)$ part of $ARIMA(p,d,q)$ model, and the computation complexity of both model fitting and prediction will be dramatically reduced. In experiment, high accuracy of processor load prediction is achieved.

Based on the analysis result we develop a fully adaptive DVS framework and a set of algorithms to adjust supply voltage online. More than 50% energy reduction is achieved when streaming CIF sequences with deadline miss rate below 10%

7. ACKNOWLEDGEMENT

This work is partially supported by the National Natural Science Foundation of China under grant No. 90207001.

REFERENCE

[1] Fabien Quilleré, Sanjay Rajopadhye , Optimizing memory usage in the polyhedral model September 2000 *ACM Transactions on Programming Languages and Systems (TOPLAS)*,

[2] A. Varma, B. Ganesh, M. Sen, S. R. Choudhary, L. Srinivasan and B. Jacob, "A Control-Theoretic Approach to Dynamic Voltage Scaling", *International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES '03)*, San Jose, CA, Oct. 2003.

[3] P. Pillai and K. G. Shin. Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems. In Proc. of the 18th ACM Symp. on Operating Systems Principles, 2001.

[4] J. Lorch and A. Smith. Improving dynamic voltage scaling algorithms with pace. In *Sigmetrics 2001*, Cambridge, MA, June 2001.

[5] http://www.aaeon.com/w3/product/aei/SOM/ESM-2615.htm