Power Prediction of Mobile Processors based on Statistical Analysis of Performance Monitoring Events

Hee-Sung Yun and Sang-Jeong Lee

Dept. of Computer Science and Engineering, Soonchunhyang Univ., Korea

Abstract

In mobile systems, energy efficiency is critical to extend battery life. Therefore, power consumption should be taken into account to develop software in addition to performance. Efficient software design in power and performance is possible if accurate power prediction is accomplished during the execution of software.

In this paper, power estimation model is developed using statistical analysis. The proposed model analyzes processor behavior quantitatively using the data of performance monitoring events and power consumption collected by executing various benchmark programs. And then representative hardware events on power consumption are selected using hierarchical clustering. The power prediction model is established by regression analysis in which the selected events are independent variables and power is a response variable. The proposed model is applied to a PXA320 mobile processor based on Intel XScale architecture and shows average estimation error within 4% of the actual measured power consumption of the processor.

Keyword: power prediction, mobile processor, performance monitoring events, regression analysis, hierarchical clustering

Initial Performance Monitoring Events of a PXA320 processor

In order to predict power for PXA320, ten performance monitoring events are selected as initial events (Table 1). The events are collected every 10ms.

Table 1. Initial Performance Monitoring Events

Event	Description
INST	The number of instructions retired
STALL	The number of stalls due to data dependency
I_TLB_MISS	The number of instruction TLB misses
D_TLB_MISS	The number of data TLB misses
I$_MISS	The number of instruction cache misses
L2$_MISS	The number of L2 cache misses
BR_MISS	The number of branch prediction misses
MULT	The number of multiplier in use
CO_STALL	The number of coprocessor stalled
BUS	The number of data bus transaction

Methodology to Generate Power Prediction Model

Figure 1 shows the process for generating power prediction model.

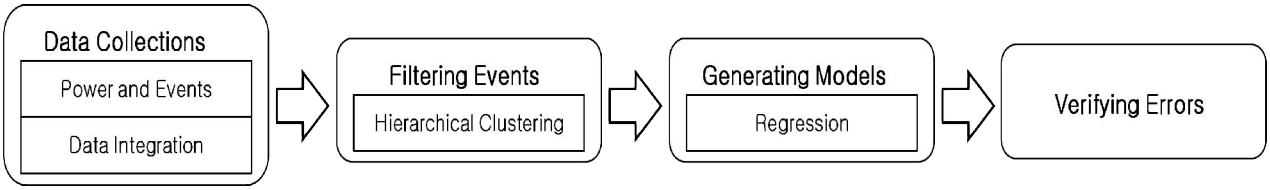


Figure 1. The Process for Generating Power Prediction Model

Filtering Performance Monitoring Evemts

In PXA320 processor, four performance events can be measured concurrently a a time. In order to find four best predictors among ten events, we cluster four groups according to their power contribution using hierarchical clustering. Figure 2 shows the result of hierarchical clustering.

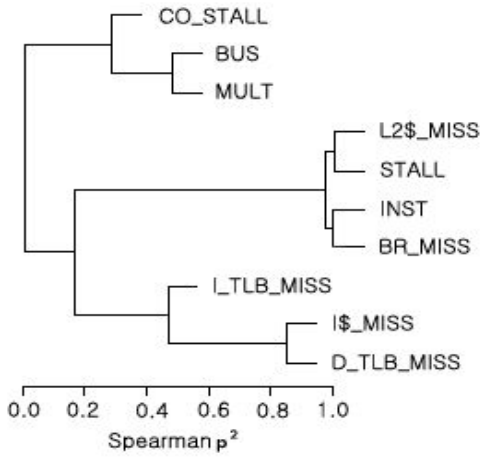
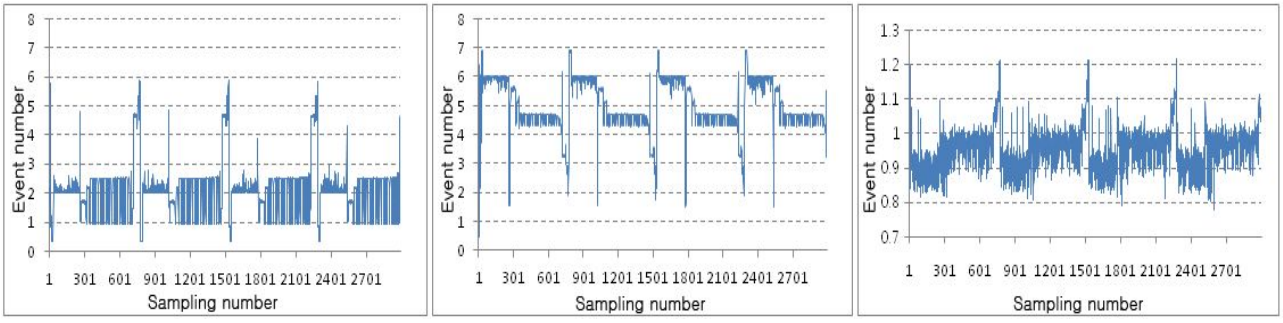


Figure 2. The Result of Hierarchical Clustering

With the result of hierarchical clustering, we selected following four events for power prediction model by excluding the events in the same group.

{ INST, STALL, I$_MISS, BUS }

Figure 3 show the traces of power consumption, STALL and INST events in bzip2 benchmark



(a) INST (b) STALL (c) Power

Figure 3. The Traces of Power Consumption, STALL and INST Events in Bzip2 benchmark

Generating Models

The power prediction model is established by regression analysis in which the selected events are independent variables and power is a response variable. Table 2 shows regression coefficients and R^2 values for each clock frequencies.

$$\boldsymbol{Power = \beta_0 + \beta_1 INST + \beta_2 STALL + \beta_3 I\$\_MISS + \beta_4 BUS}$$

Table 2. Regression Coefficients and R^2 Values

Frequency	β_0	β_1	β_2	β_3	β_4	R^2
806MHz	$9.561 * 10^{-1}$	$1.657 * 10^{-8}$	$-7.324 * 10^{-7}$	$-1.778 * 10^{-8}$	$2.402 * 10^{-6}$	0.967
624MHz	$6.316 * 10^{-1}$	$3.486 * 10^{-8}$	$-5.839 * 10^{-9}$	$3.086 * 10^{-7}$	$1.677 * 10^{-6}$	0.973
403MHz	$2.781 * 10^{-1}$	$1.511 * 10^{-8}$	$-3.566 * 10^{-9}$	$1.603 * 10^{-7}$	$1.290 * 10^{-6}$	0.935

Figure 4 is the result of residual analysis for the above regression model to show the distribution of 속 difference between the predicted values and the observed values. In the figure we can verify that the residual is randomly distributed around zero without bias.

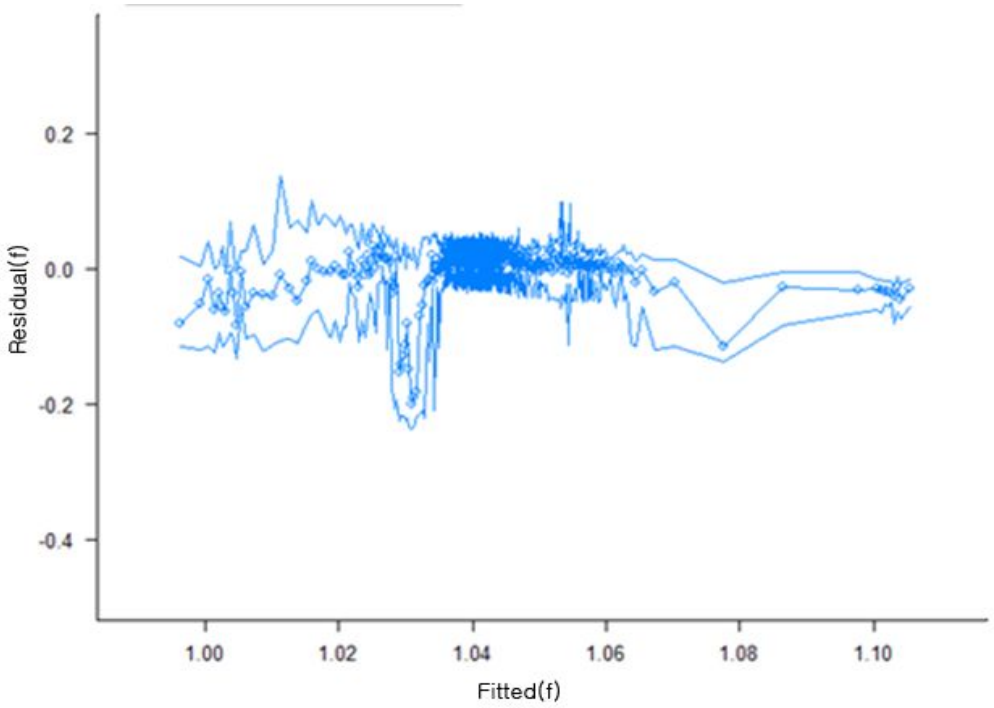


Figure 4. The Result of Residual Analysis

Experimentation Environment

Table 3 shows ten benchmark programs which are consisted of MiBench and SPEC CPU2006 benchmarks. We use different input data among learning phase with regression and verification phase for the model. For the input data of MiBench we use 20 data from MiDataSets [16][17].

Table 3. Benchmark Programs

Program	Type	Description	Learning Data	Verification Data
Jpeg	MiBench	JPEG Decoder	Odd MiDataSets	Even MiDataSets
Gsm	MiBench	GSM Decoder	Odd MiDataSets	Even MiDataSets
Dijkstra	MiBench	Shortest Path Computation	Odd MiDataSets	Even MiDataSets
Ispell	MiBench	Spelling Checker	Odd MiDataSets	Even MiDataSets
bzip2	SPEC	File Compression	dryer.jpg	input.program
Gobmk	SPEC	Go Game	dniwog.tst	connect.tst
Blowfish	MIBench	Blowfish Encryption	X	MiDataSets
Adpcm	MIBench	ADPCM Decoder	X	MiDataSets
Perlbench	SPEC	Perl Interpreter	X	splitmail.pl
Mcf	SPEC	Vehicle Scheduling	X	inp.in

In order to collect power consumption of the processor, we measure current and voltage every 10ms with NI digital meters (Figure 5).

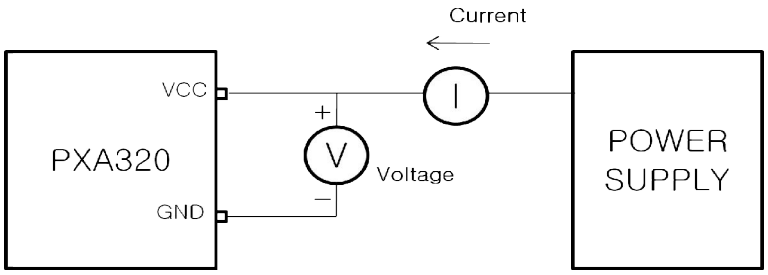


Figure 5. The Measurement of Current and Voltage for PXA320 Processor

Figure 6, 7 shows the experimentation setup. In order to collect performance monitoring events and synchronize with power measurement of digital meters, we implemented loadable modules in Linux.

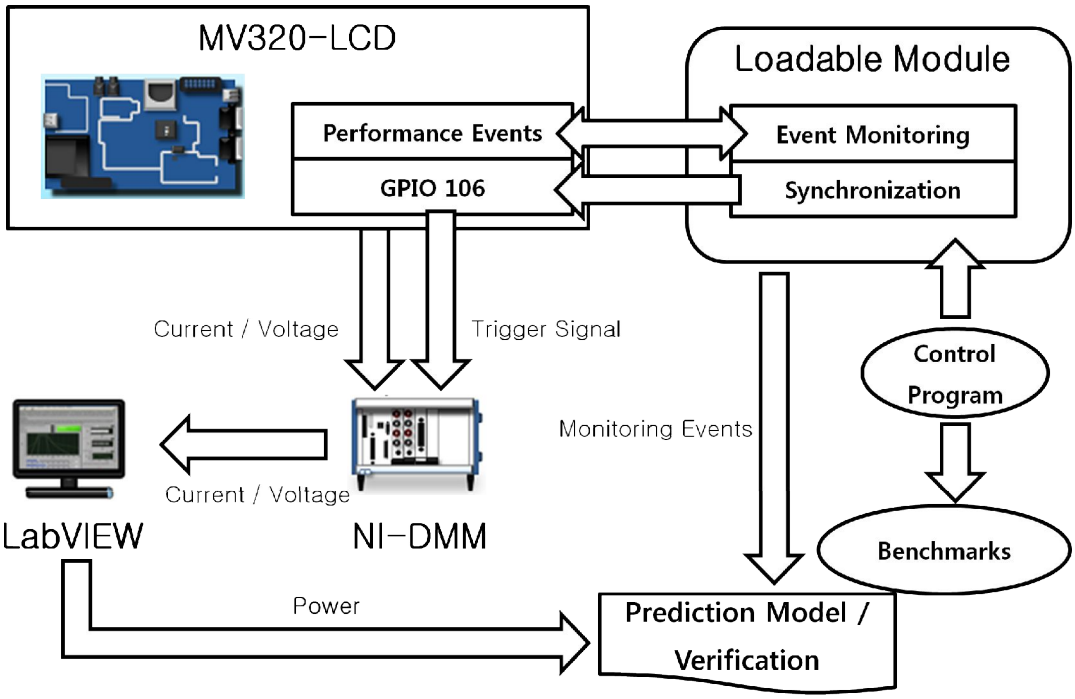


Figure 6. Experimentation Setup

Figure 7. Experimentation Photos

Experimentation Results

Figure 8 shows the predicted and measured values of power, and figure 9 shows prediction errors for each clock frequencies. They show average estimation error within 4% of the actual measured power consumption of the processor. Figure 10 shows error rate when sampled intervals are 10ms, 50ms, 100ms respectively. It shows the prediction accuracy is not affected by sampling interval length.

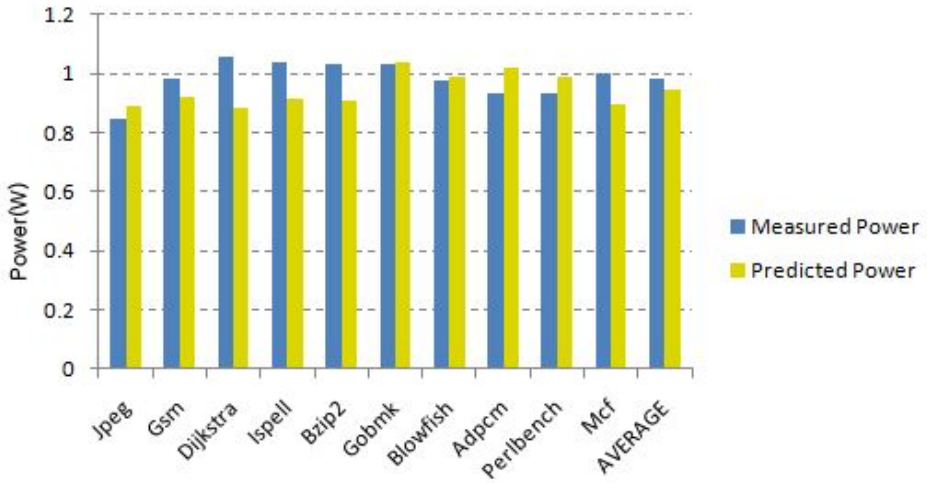


Figure 8. The Measured Power and Predicted Power

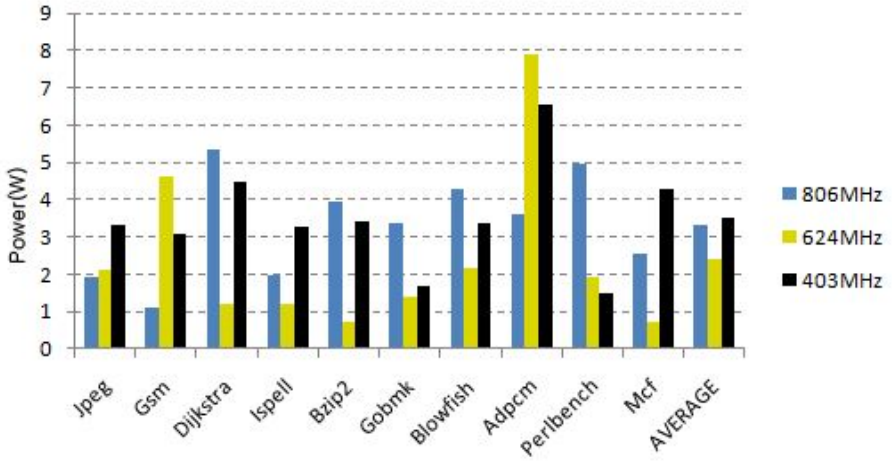


Figure 9. Prediction Error

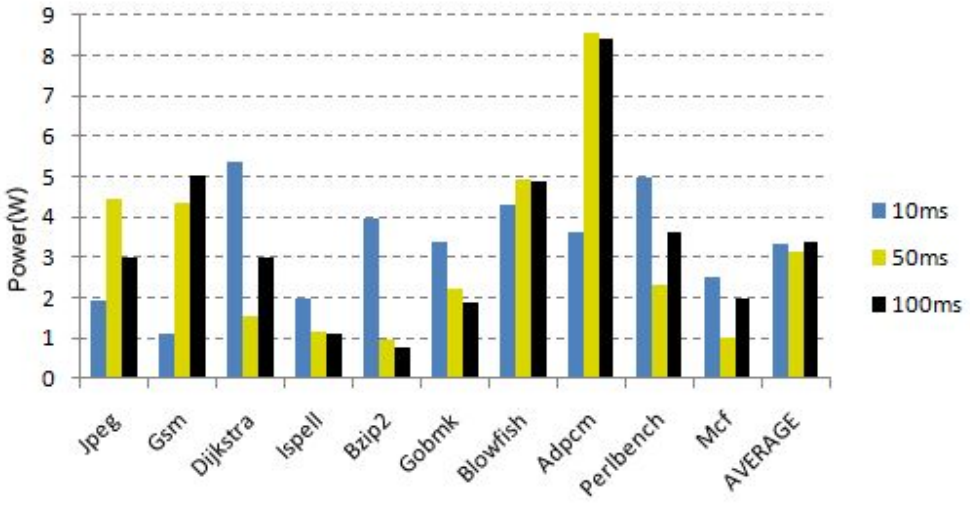


Figure 10. Prediction Error with Variation of Sampling Intervals

References

[1] B. Sprunt, "Basics of Performance-Monitoring Hardware," *IEEE Micro Vol.22 Issue 4*, pp.64-71, July-August 2002.

[2] G. Contreras and M. Martonosi, "Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events, " *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED'05)*, Aug. 2005.

[3] K. Rajamani et al., "Application-Aware Power Management", *Proc. Int'l Symp. Workload Characterization (IISWC '06)*, Oct. 2006

[4] Q. Wu et al., "Dynamic Compilation Framework for Controlling Microprocessor Energy and Performance," *Proc. Int'l Symp. Microarchitecture (MICRO '05)*, Nov. 2005

[5] C. Isci and M. Martonosi, "Phase Characterization for Power: Evaluating Control-Flow-Based and Event-Counter-Based Techniques," *Proc. Int'l Symp. High-Performance Computer Architecture (HPCA '06)*, Feb. 2006.

[6] C. Isci, G. Contreras, and M. Martonosi, "Live, Runtime Phase Monitoring and Prediction on Real Systems with Application to Dynamic Power Management", *Proc. Int'l Symp. Microarchitecture (MICRO '06)*, Dec. 2006

[7] H. Sasaki, Y. Ikeda, M. Kondo and H.Nakamura, "An Intra-Task DVFS Technique based on Statistical Analysis of Hardware Events", *Proc. Int'l Conf. Computing Frontiers*, May 2007

[8] S. Lee, H. Lee and P. Yew, "Runtime Performance Projection Model for Dynamic Power Management," *Proc. Asia-Pacific Computer Systems Architecture Conference (ACSAC '07)*, pp.186-197, Aug. 2007.

[9] B. Lee and D. Brooks, "Accurate and Efficient Regression Modeling for Microarchitectural Performance and Power Prediction," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '06)*, Oct. 2006.

[10] B. Lee and David M. Brooks. "Spatial Sampling and Regression," *IEEE Micro Vol. 7 Issue. 3*, pp.74-93, May-June 2007

[11] W. Mathur and J. Cook, "Improved Estimation for Software Multiplexing of Performance Counters", Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005

[12] 3rd Generation Intel XScale® Microarchitecture Developer's Manual, Intel, 2007

[13] Monahans P Processor System and Timer Configuration Developers Manual, Marvell, 2006

[14] M. Guthaus et al., "MiBench: A Free, Commercially Representative Embedded Benchmark Suite," *Proc. Annual Workshop on Workload Characterization*, Dec. 2001.

[15] MiBench Version 1.0, http://www.eecs.umich.edu/mibench/

[16] G. Fursin et al., "MiDataSets: Creating The Conditions For A More Realistic Evaluation of Iterative Optimization," *Proc. Int'l Conf. High Performance Embedded Architectures & Compilers (HiPEAC '07)*, Jan. 2007

[17] MiDataSets for MiBench, http://midatasets.sourceforge.net/

[18] Standard Performance Evaluation Corporation (SPEC), http://spec.org/

[19] R Project for Statistical Computing, http://www.r-project.org/

[20] MicroVision Co., http://mvtool.co.kr

[21] National Instruments LabVIEW, http://www.ni.com/labview