ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Run Time Power Optimization in Multicore Embedded Systems

Y.Ramanamma[1]

M. Tech, Dept. of ECE, JNTUK, Kakinada, India[1]

ABSTRACT: As the wireless communication technology sophisticated, much more multimedia applications are boosted in new generation's smart phones, though these features allow dynamic power consumption to huge values, the decreasing size and reducing battery life time keep limiting the available power resources, thus the need of efficient power management techniques invoked. Though the power consumption is static and dynamic the focus of this paper is dynamic or at runtime of the system DPM technique can selectively turnoff the devices while they are idle partially used while increasing switching power and latency and then DVFS can decrease latency still reducing performance thus DPM and DVFS can meet both ends still reducing performance. To address this we propose an alternative that uses reinforcement learning to explore the trade-off between power saving and performance using DVFS and DPM with AVS(Adaptive Voltage Scaling) implementation as Smart Power Manager(SPM) as a Linux LKM on Sitara ARM 572x EVM.

KEYWORDS:RTM(Run Time manger),DPM (Dynamic power management),DVFS (Dynamic Voltage and Frequency Scaling), EWMA(Exponential Weighted Moving Average),Q-Learning.

I. INTRODUCTION

Modern embedded systems execute applications, which interact with the operating system and hardware differently depending on the type of work load this will give widely varying power consumptions at run time though there are many system level power management techniques available for run time power optimization such as Linux default ondemand governor,which is switching between two frequencies highest and lowest which are available on the Soc in this paper we are implementing a new technique which can be switched between all available OPPs on the Soc dynamically and percentage power consumption is increased using this methodology when compared to Linux ondemand governor and our method is implemented as a Linux loadable kernel module which is having two algorithms prediction and decision algorithm here prediction algorithm is EWMA for CPU work load prediction and Decision algorithm is for setting voltage and frequency That is OPP on PMIC using the system calls and cpufreq API .

II. RELATED WORK

Dynamic Power management techniques

Dynamic power management techniques available in atypical embedded system are cpuidle,DVFS and smart reflex technology in DVFS method it uses different OPPs according to the application requirement such as performance which will be run at highest frequency when performance is the main criteria, power save which will be run at lower frequency when power save is the main criteria and ondemand which is the default governor in embedded systems can be switched between highest and lowest frequencies that are available on the SoC but these techniques are not giving significant power saving therefore we are going for new methodology which we are implemented as Run Time Manager

Problem formulation

There are three dependencies that are relevant to this work – the sub-threshold leakage power of a system and its lifetime reliability are both dependent on temperature; the dynamic power consumption is dependent on the voltage and frequency of operation; and the temperature is dependent on the voltage, frequency and also on the application workload. In this section, we provide an overview of these dependencies and formulate the objective we optimize in

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

this work.Power Consumption As discussed in [He et al. 2004], a typical processor can be considered as being in one of the three power states Standby Mode in this mode, a processor is idle. The total power consumption is comprised of the leakage component (Ps) only; — Active Mode: in this mode, a processor is active and executes instructions and the total power is comprised of leakage (Ps) and dynamic (Pd) components; and in Inactive Mode in this mode, a processor is usually power gated. The total power is comprised of the reduced leakage component (Pr). The dynamic power of a processor is directly proportional to the frequency (f) of operation and quadratically proportional to the voltage (V), i.e., $Pd \propto f \propto V^{2}$\

III. PROPOSED METHOD

Run Time Manager proposed

This RTM consisting of two algorithms 1)EWMA for prediction of CPU work load and 2)Decision making algorithm ,which is a variant of reinforcement algorithm (Q-learning algorithm), which will take decisions according to the performance requirement of currently executing application. This algorithm is implemented as a look up table consisting of state and actions. CPU work load is taken as state space and voltage frequency pair is taken as action space. It will take decisions according to the performance requirement of the application and set the required OPP using cpufreq API , which is available on the processor this the action of Q-learning algorithm.

EWMA Algorithm

The Prediction Unit estimates the workload for the next frame using a modified form of EWMA. The EWMA algorithm is widely used in the literature because of its lightweight implementation. The predictor works as an infinite impulse response filter that generates a prediction of the future value based on the average of the previous values weighted exponentially, where the most recent values have greaterweights than the older ones. This is shown as: $x(n + 1) = x(n) * \lambda + \bar{x}(n) * (1 - \lambda)$ where $0 \leq \lambda \leq 1$.

Decision algorithm

Decision algorithm is a variant of reinforcement learning algorithm i.e Q-learning algorithm, which is consisting of three phases Exploration, learnig phase and exploaitation phase which is the final and the algorithm will take decisions according to the performance requirement of application based on calculated reward from the Q-table.

Power Modelling

For dynamic power calculation we have used hardware performance counters in the cortex A series processors such as Data cache access, cycle count the below figure shows power modeling model for dynamic power calculation

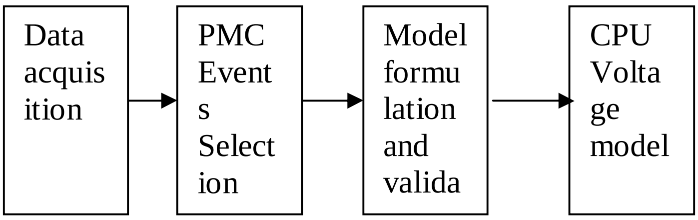

Figure 1 Power modeling using PMC events

The above figure shows power modeling of PMC events for dynamic power modeling of embedded systems. We have taken embedded Mibench applications for power modeling of our system for correlation coefficients the below fig1 shows correlation factor variation for susan Mibench application where

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

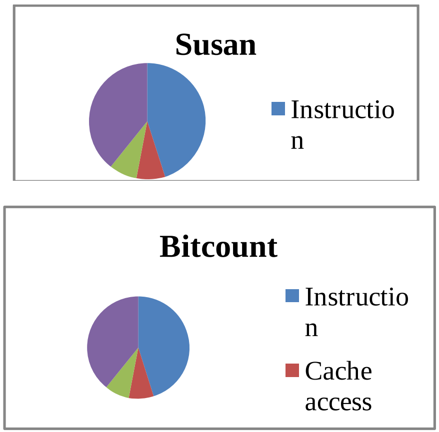

Figure 2 Correlation factor variation for mibench applications

IV. RESULTS

Prediction using EWMA algorithm gives 2.3% error when compared to measured work load. We have run three different videos of315,640 and 725 then we have observed that our methodology has improved on average 6.6% power when compared to Linux default on demand governor but reducing performance by 0.9%, which is not reducing the overall performance of the application. Fig 2indicates predicted work load and current work load of CPU and fig3 shows performance of sdefault on demand governor and RTM where in our proposed method power consumption reduces when compared to on demand.

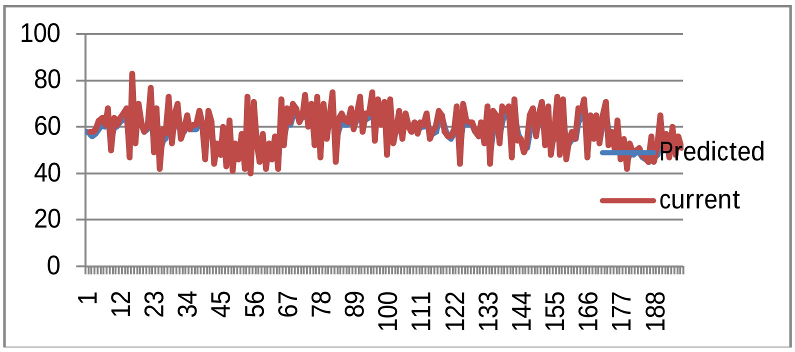

Figure 3 Predicted and current work load

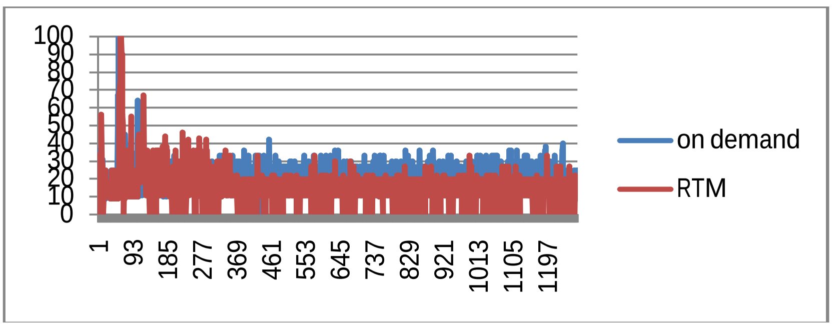

Figure 4 Performance of ondemand And RTM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

V. CONCLUSION

A low overhead hierarchical run time manager is proposed for multi core embedded systems for energy efficiency continuing work includes Kernel implementation of the run-time manager, extending the approach to include heterogeneous components such as GPUs and FPGAs.

REFERENCES

[1] Anup Das., Geoff V.Merrett., Bashir M.Al-Hashin., A. (2015). “Hardware software interaction for run time power optimization”. IEEE.

[2] Anup Das., Geoff V.Merrett.,”Work load change point detection for run time temperature management”ss ,2015,IEEE

[3] Anup Das., Bashir M. Al-Hashimi., Geoff V. Merrett., 2015,Adaptive and Hierarchical Run-time Manager for Energy-Aware Thermal Management of Embedded Systems” ACM Transactions on Embedded Computing System

[4] Anup Das., Akash Kumar., Bharadwaj Veeravalli., Rishad Shafik., Geoff Merrett., Bashir Al-Hashimi., School of ECS, University of Southampton and Department of ECE, National University of Singapore. ACM Transactions on Embedded Computing System

[5] Yung-Hsiang Lu., Tajana Simuni., Giovanni De Micheli .,Computer System Laboratory, Stanford University.s “Software Controlled Power Management”

[6]Ying Tan Parth., Malani Qinru., Qiu Qing.,Wu Department of Electrical and Computer Engineering State University of New York at Binghamton.”Workload Prediction and Dynamic Voltage Scaling for MPEG Decoding”.

[7] Rishad A. Shafik, Member, IEEE, Sheng Yang, Member, IEEE, Anup Das, Member, IEEE, Luis A. MaedaNunez, Member, IEEE, Geoff V. Merrett, Member, IEEE & Bashir M. Al-Hashimi, Fellow, IEEE.”Learning Transfer-based Adaptive Energy Minimization in Embedded Systems”. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[8] Hwisung Jung ., Massoud Pedram ., University of Southern California Department of Electrical Engineering.” Continuous Frequency Adjustment Technique Based on Dynamic Workload Prediction”.

[9] H. Jung ., M. Pedram, “Supervised Learning Based Power Management for Multicore Processors,” IEEE TCAD, 2010.

[10] “Perfmon: Linux Profiling with Performance Counters,” Downloadable software with documentation, http://www. hpl. hp. com/research/linux/perfmon, 2012.

[11] F. Sironi, M. Maggio, R. Cattaneo, G.F. Del Nero, D. Sciuto, and M.D. Santambrogio. 2013. “ThermOS: System Support for Dynamic Thermal Management of Chip Multi-processors. In International Conference on Parallel Architectures and Compilation Techniques”.

[12] C. Bienia, S. Kumar, and K. Li. 2008. “PARSEC vs. SPLASH-2: A Quantitative Comparison of Two Multithreaded Benchmark Suites on Chip-Multiprocessors”. In IEEE Symposium on Workload Characterization