International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2010

A Novel Embedded Statistical Scheduler for Realising High System-QoS

Suman Paul, A. K. Jana, M. K. Pandit
Electronics & Communication Engg, Dept, Haldia Institute of Technology (HIT)
C/o ICARE (Indian Centre for Advancement of Research and Education)
Haldia -721657 WB INDIA

`mkp10011@yahoo.com`

ABSTRACT—*Development of schedulers for real-time embedded systems is an emerging area of research as it addresses timely scheduling issues like co-operative scheduling, pre-emptive scheduling, micro-kernel based scheduling, etc. Use of statistical schedulers over deterministic schedulers offers the advantage of improved bandwidth utilization, better delay and resource management, higher throughput and thus higher system QoS (quality of service). We report here for the first time a novel stochastic scheduling method based on optimized Markov chains where the pre-scheduled Markov transition probabilities are subsequently dynamically tuned by the central scheduler to give the best system QoS.*

KEYWORDS— *QoS, Schedulers, Embedded system, Markov proces , Probability of Error*

1. INTRODUTION

Optimizing the CPU usage times allocated to processes for providing high QoS is a key issue and challenge in modern embedded system design. In a multi-tasking environment where an embedded processor runs several processes, the CPU usage time provisioned for each process is analytically intractable and needs simulation study. In our simulation we consider **Markov statistics** to get the initial estimates for the scheduled times. They are later dynamically tuned by the scheduler using **feedback control** for maintaining minimum aggregate probability of error of the system and thus best system quality of service (QoS).

2. RELATED WORK

Cazorla et al [1] have studied QoS for embedded systems. But the QoS considered is specific for an individual process. Peha et al [2] suggested some new approaches towards scheduling but they are not **QoS-aware**. Matschulat et al [3] and Tomoyoshi et al [4] have studied QoS [5-7] in embedded systems but none of them considered Markov chains. The concept of system QoS based on Markov statistics, is for the first time we have worked out here to our best knowledge.

Most of the earlier work [3-6] on finite state modelling of schedulers considered the states of each process as of four types: 1) sleeping 2) ready 3) executing and 4) blocked. This is not an enabling technique to calculate system QoS. We, on the other hand, model here each of the processes in a multi-tasking environment as an individual Markov process and calculate the system QoS. To our best of knowledge, this is the **first such approach** [8,9].

3. METHOLODGY

This paper reports a Finite-state machine (FSM) based Markov chain model for schedulers where each process is modelled as a Markov state. The processes settle to a **steady** state

DOI : 10.5121/ijcsit.2010.2606

probability distribution as time evolves. Fig. 1, Fig. 2. and Fig. 3. illustrate **convergence** of two states to steady state equilibrium with initial state probability distributions $\prod_0$ = [0.50:0.50], [0.40:0.60] and [0.10:0.90], respectively, considering 100 iterations for each case. Thus irrespective of change of initial considerations of all these cases the two states converge to a **steady state** value, which justifies our consideration of Markov process.

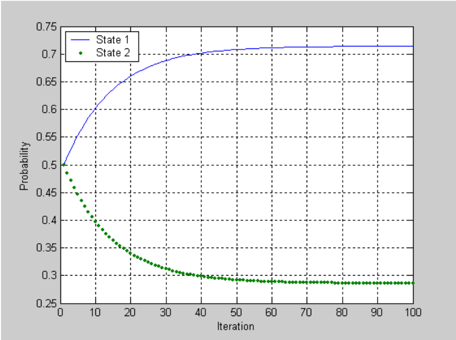

Fig. 1. Convergence of two states having initial state distribution [0.50: 0.50]

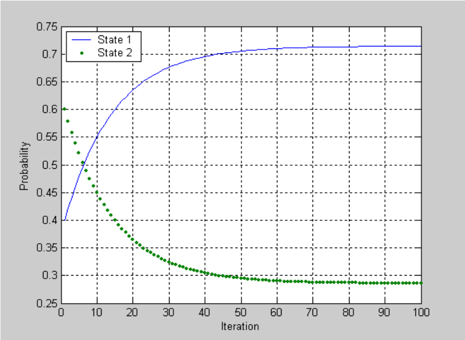

Fig. 2. Convergence of two states having initial state distribution [0.40: 0.60]

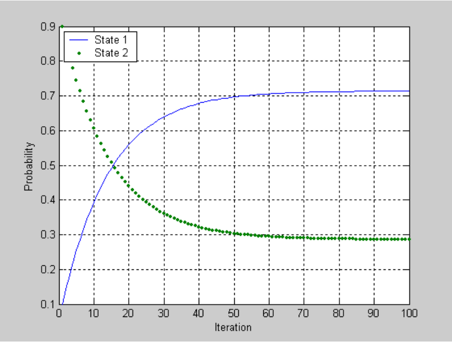

Fig. 3. Convergence of two states having initial state distribution [0.10 : 0.90]

Assume that the processor is working on process (state) k at time t and that both the next process (state) and the corresponding component of the error vector, e_{k+1} are to be simulated.

The first step is to generate two **uniformly distributed** random variables, U_1 and U_2. The first random variable, U_1, is used to determine the new process. After the new process is established, the occurrence of error is checked by comparing U_2 with a threshold given by the corresponding element of the error-generation matrix B.

4. OUR MODEL - FINITE STATE MACHINE (FSM)

In our model we consider three processes (which are adequate for many low-end embedded applications) assigned to a single processor. Considering a finite state machine (FSM), the three states $<P_1>$,$< P_2>$,$< P_3>$ define the execution of three corresponding processes by the CPU. As in a single processor system the CPU executes only one process at a time, in order to execute several processes, transition from one process to another via **context switching** is essential. Context switching is expensive because it requires CPU overhead so it cannot be allowed at arbitrary times. It must be kept to a minimum by proper allocation of the state transition probabilities p_{ij}.

Fig. 4 demonstrates the three-process state transition diagram. The directed edges determine CPU‘s transition or switching from one process to another. The CPU may stay at its present state or switch over to any one of the other two states, determined by external **events**. The weights of the corresponding edges are considered as the transition probabilities.

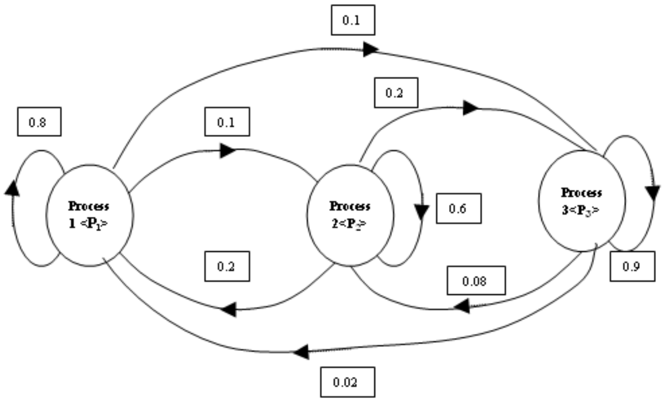

Fig. 4. State transition model for three processes

5. SIMULATION METHOD

For simulation purpose, we consider two matrices 1) A, a state transition matrix for the Markov model assumed (here three-state model) and 2) B, an error probability matrix.

$$A = \begin{bmatrix} 0.80 & 0.10 & 0.10 \\ 0.20 & 0.60 & 0.20 \\ 0.02 & 0.08 & 0.90 \end{bmatrix} \quad B = \begin{bmatrix} 0.9990 & 0.9500 & 0.9900 \\ & & \\ 0.0010 & 0.0500 & 0.0100 \end{bmatrix} \qquad (1)$$

$$A_{tun} = \begin{bmatrix} 0.80\overset{\sim}{+}2\Delta_1 & 0.10+\Delta_1 & 0.10+\Delta_1 \\ 0.20+\Delta_2 & 0.60-2\Delta_2 & 0.20+\Delta_2 \\ 0.02+\Delta_3 & 0.08+\Delta_3 & 0.90-2\Delta_3 \end{bmatrix} \qquad (2)$$

The simulation is done by MATLAB version 7.0. We perform simulation for calculating error vector which give error positions in 20000 sequences (iterations) considering random numbers, and the state sequence matrix which deals with state transitions. The probability of finding the processor in a given state can be calculated from the state sequence matrix. Similarly error probability can be found from the error vector.

6. RESULTS

We find the steady-state probabilities of processes 1, 2, and 3 as 0.2432, 0.1634 and 0.5934 respectively and the error probability as 0.0097. We take different values of sequences starting from 1000 to 3000000 and simulate probability of error (P_e) for each case and plot the results in Fig 5. The results indicate that with a high count of iterations, P_e settles to a steady state value. This **validates** our Markov model.

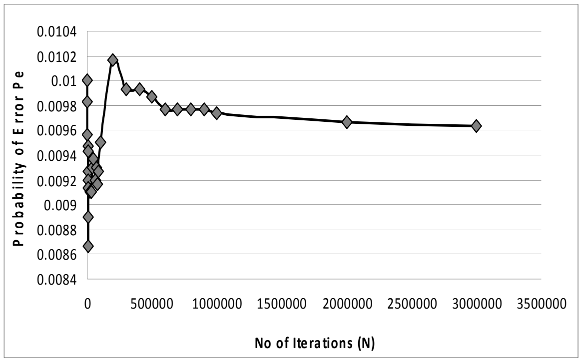

Fig. 5. Pe converges to a steady state

In practice, the CPU usage allocation to each process is not static over time and depends on the **events**. The scheduler **dynamically** monitors the system QoS and takes corrective action using a feedback control loop as shown in Fig 6.

To find minimum P_e and thus optimize system performance,

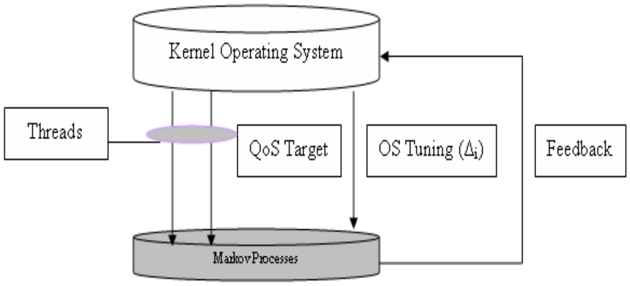

Fig. 6. Feedback control system for our scheduler

the feedback loop modifies A with tuning parameters Δ_1 , Δ_2 and Δ_3 as given in Equation 2. Their values are kept in the range of 0 to ±0.2. for a given error-generation matrix B.

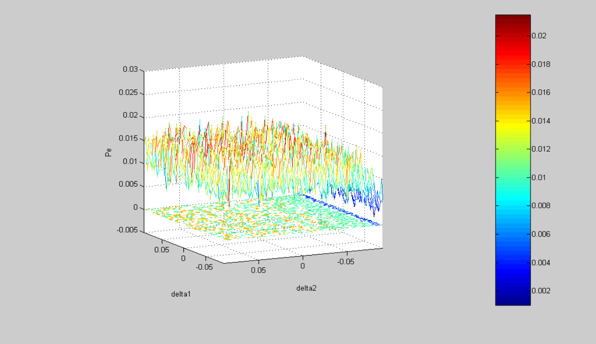

Fig. 7.. Pe vs Δ_1, Δ_2 ($\Delta_3 = 0$)

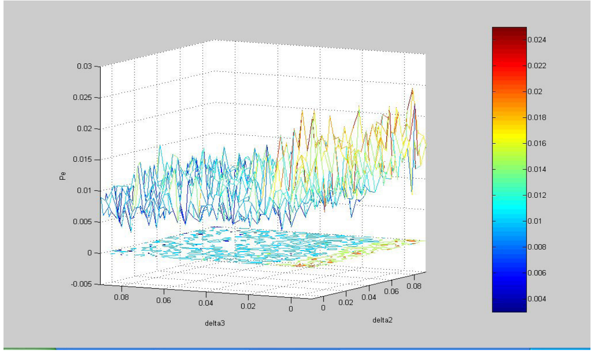

Fig. 8. Pe vs Δ_2, Δ_3 ($\Delta_1 = 0$)

P_e is 3D-contour plotted as function of Δ's in Figs 7-9. P_e is seen to be **globally minimum** at 0.001 if values of Δ_1 , Δ_2 and Δ_3 are kept 0.025, -0.09 and 0, respectively. These values are expected because with this combination the second process has almost same state transition probabilities as the first process which has the lowest probability of error at 0.001. This **confirms** the **correctness** of our model. This globally minimum P_e gives the **highest system** QoS, resulting in **best** system performance.

It is to be noted that in practice, implementation of exact values of Δ's is determined by the availability of idle periods during machine cycle operations.

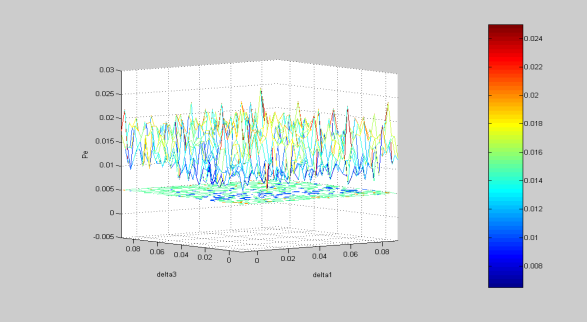

Fig. 9. Pe vs Δ_1, Δ_3 ($\Delta_2 = 0$)

7. CONCLUSION

We have proposed and designed a novel scheduler using Markov statistics for use in embedded systems. The Markov model presented here deals with central scheduling but it can be extended to take account of co-operative scheduling. This can be realised by inter-process communication.

ACKNOWLEDGEMENT

We sincerely acknowledge financial support from the TEQIP, Govt of India (Technical Education Quality Improvement Programme) – Phase 1 grant.

REFERENCES

[1] F. J. Cazorla, Peter M.W. Knijnenburg and E. Fernández, “QOS for high-performance SMT processors in embedded systems”, IEEE Micro, July-Aug 2004, pp. 24-31.

[2] J. M. Peha and F. A. Tobagi, “Cost-Based Scheduling and Dropping Algorithms To Support Integrated Services”, IEEE Transactions on Communications, Vol. 44, No. 2, Feb. 1996, pp. 192-202.

[3] D. Matschulat, C. A. M. Marcon, F. Hessel, ’A QoS scheduler for Real-time Embedded Systems’, 9th Int symposium on Quality Electronic Design (Isqed-2008), March17-19, 2008

[4] S. Tomoyoshi, T. Kosuke, ”Table-based QoS control for embedded real-time systems”, Proc of the ACM SIGPLAN 1999 workshop on Languages, compilers, and tools for embedded systems, 1999, pp. 65-72.

[5] R. Jain, C. J. Hughes and S. V. Adve, ”Soft Real-time Scheduling on Simultaneous multithreaded processors”, Proc of 23rd Real-time systems symposium (RTSS-23), IEEE Press, 2002, pp. 134-135.

[6] A. Snaverly, D. M. Tullsen and G. Voelker, “Symbiotic job scheduling with priorities for a simultaneous multithreaded processor”, Proc of 9th International Conf on Architectural Support for programming languages and operating systems (ASPLOS-9), ACM Press, 2000, pp. 234-244.

[7] S. Addepallil, P. Andersen and G. L. Barnes, “Efficient Resource Matching in Heterogeneous Grid Using Resource Vector”, Int Journal of Computer Science and Information Technology, Vol. 2, No. 3, June 2010, pp. 1-10.

[8] S. Paul, A. K. Jana and M. K. Pandit, “A Novel Embedded Scheduler Based on Markov Statistics for Generating High QoS”, International Conf on Communication and Computing Intelligence (INCOCCI 2010), Kongu Engineering College, Tamilnadu, 27-29 Dec 2010.

[9] B. Bag, A. K. Jana and M. K. Pandit, “A Novel Reconfigurable Video Burst Processor Offering High QoS”, International Springer Conf on Advances in Comm Networks and Computing”, CNC 2011, Bangalore, Feb 23-24, 2011.

Brief Biography

Suman Paul

Suman Paul received his bachelor’s and master’s degree in electronics engg and computer science, respectively in 2005 and 2008, respectively. He is working now towards PhD degree in the field of embedded systems. He worked as associated researcher in the prestigious Indian Institute of Management (IIM), Joka in 2007. He has published 4 international research papers.

Asim K. Jana

Asim K. Jana received his bachelor's and master's degree in electronics Engg from Electronics and Telecom Engg Dept, Jadavpur University, India in 1988 and 1995, respectively. He has 10 years of industrial experience in the domain of embedded systems. Currently he is an associate professor on computer systems. He has 10 international research papers.

M K Pandit

Malay K Pandit received his B.E and M. E degrees in Electronics Engineering from Electronics and Telecom Engg Dept, Jadavpur University, India in 1989 and 1991, respectively. He received his PhD from UK's renowned Cambridge University in 1996. He did his post-doc from the Optoelectronics Research Centre, City University of Hong Kong till 2002 where he pioneered the use of polymers for optical waveguide applications. .He then took a corporate career where he worked in a fibre optic company "FONS (I) Ltd" in the domain of optical networking. Now he is a full Professor in the Electronics Engg Dept of the Haldia Institute of Technology where he focuses on embedded systems, including their usage in WDM optical networks. He has 45 international publications in this area.

Dr. Pandit was awarded the National Scholarship of the Government of India in 1983 and the Nehru Scholarship of the Government of India during his Ph.D. studies.