Energy-Aware Online Algorithm to Satisfy Sampling Rates with Guaranteed Probability for Sensor Applications*

Meikang Qiu[1] and Edwin H.-M. Sha[2]

[1] University of New Orleans,
New Orleans, LA 70148, USA
mqiu@uno.edu
[2] University of Texas at Dallas,
Richardson, TX 75083, USA
edsha@utdallas.edu

Abstract. Energy consumption is a major factor that limits the performance of sensor applications. Sensor nodes have varying sampling rates since they face continuously changing environments. In this paper, the sampling rate is modeled as a random variable, which is estimated over a finite time window. We presents an online algorithm to minimize the total energy consumption while satisfying sampling rate with guaranteed probability. An efficient algorithm, EOSP (*Energy-aware Online algorithm to satisfy Sampling rates with guaranteed Probability*), is proposed. Our approach can adapt the architecture accordingly to save energy. Experimental results demonstrate the effectiveness of our approach.

1 Introduction

Sensor networks are emerging as a main technology for many applications, such as national security and health care. In this context, a key problem that ought to be tackled is that of devising embedded software and hardware architectures that can effectively operate in continuously changing, hard-to-predict conditions [1].

In sensor network applications, such as moving objects tracking, tasks may not have fixed sampling rate. The sampling rate may vary, depending on the rate the object moves. At same time, the time window of sensor systems is fix. Hence, how to select the proper number of *processing engine* PEs and adopt different sampling rates becomes an important problem. The existing methods are not able to deal with the uncertainty of sampling rate. Therefore, either worst-case or average-case sampling rates for these tasks are usually assumed. Such assumptions, however, may not be applicable for the hard-to-predict conditions and may result in an inefficient task assignment. In this paper, we models the sampling rate in each time window as a random variable [2].

* This work is partially supported by TI University Program, NSF EIA-0103709, Texas ARP 009741-0028-2001, NSF CCR-0309461, NSF IIS-0513669, and Microsoft, USA.

R. Perrott et al. (Eds.): HPCC 2007, LNCS 4782, pp. 156–167, 2007.

In sensor applications, such as camera based sensors, same tasks can be processed under different voltage levels with different energy consumptions. DVS (Dynamic voltage scaling) is one of the most effective low power system design techniques. We want to select proper number of PEs and assign a proper voltage to each task of a sensor such that minimize the total energy consumption while satisfying the total sampling rate constraint with a guaranteed confidence probability.

This paper use adaptive approach for online customization of embedded architectures that function in non-stationary environments [1]. we design an algorithm, EOSP (*Energy-aware Online algorithm to satisfy Sampling rates with guaranteed Probability*), to select the proper number of PEs and assign different voltage levels to the PEs to minimize total energy consumption. The obtained voltages will affect the adaptation thresholds of control policies.

The experimental results show that EOSP achieves a significant reduction in total energy consumption on average. For example, with 4 voltages, compared with Method 1 (worse-case scenario without DVS), EOSP shows an average 33.8% reduction in total energy consumption while satisfying sampling rate constraint 360 with probability 0.80.

In the next section, we introduce necessary background, including basic definition and models. The algorithm is discussed in Section 3. We show our experimental results in Section 4. Related work and concluding remarks are provided in Section 5 and 6, respectively.

2 Basic Concepts and Models

In this section, we introduce some basic concepts and models which will be used in the later sections. First the sensor model is given. Next, two kinds of PE processing mode are introduced. Then we give the basic formula of DVS. Finally, we give the formal definition of VAP problem. An example is given throughout the whole section.

2.1 The Sensor Model

Sensor usually works in highly continue changing environments. For example, consider a camera based sensor, which is tracking a moving object, such as a person or vehicle [3,1]. The tracking granularity requirement demands one image sample per meter of distance traveled by the object, in order to have a trace of the object's trajectory accurate to within one meter distance of the object's actual location at all times. If the object is traveling at a speed of 20 m/s this sampling speed would translate to having 20 samples/s for the camera. We modeled the system dynamics with discrete events formulated over a fixed time window used to sample the future performance requirements of the system.

For the adaptive architecture, the working procedures are as follows. During look ahead for the next window, the incoming data is buffered. The controller uses the inputs from the sampling rate look ahead to update its control policy.

The controller makes changes to the pool of hardware with the updated policy after each window.

The time *window length* (WL) is a design parameter, and will have to be decided by the designer based on empirical data obtained from simulations of the particular application. In the tracking example the sampling rate may vary, depending on the rate the object moves. If sampling rate is high then the number of hardware resources turned on is larger in order to meet the tighter timing constraint. There is an upper limit to sampling rate based on the number and working mode of hardware processing engines (PEs) being made available for it by an architecture.

Problem Definition: For a given adaptive architecture, how to select the proper number of processing engine (PE) and assign voltage levels to the PEs for each time window of a sensor such that the total energy consumption can be minimized with a guaranteed confidence probability satisfying sampling rates.

2.2 Sampling Rate Estimator and PE Processing Method

The sampling rate is a random variable, which is decided by the moving rate of objects. We use a fixed time window to collect the data. The method we estimate the sampling rate for each window is based on the previous four average sampling rates. For example, if the average sampling rates are S_{n-1}, S_{n-2}, S_{n-3}, and S_{n-4}, for S in previous 4 time windows, then for current window, $S_n = S_{n-1} + 0.5 * (S_{n-1} - S_{n-2}) + 0.3(S_{n-2} - S_{n-3}) + 0.2(S_{n-3} - S_{n-4})$. We use several similar functions to estimate S_n. The estimated S_n is also a random variable. For example, the S shown in Figure 1 (a) is a random variable. "P1" represent the corresponding probability of every sampling rate value.

S	P1
80	0.10
60	0.10
50	0.20
40	0.20
30	0.40

(a)

S	P2
80	1.00
60	0.90
50	0.80
40	0.60
30	0.40

(b)

Fig. 1. (a) Distribution function of sampling rate. (b) Cumulative distribution function of sampling rate.

When the sampling rate is high, we need use more PEs to process the collected data. The energy consumption of each PE includes two parts.

$$E_{total} = E_{base} + E_{process} \tag{1}$$

There are two cases about PE processing modes.

1. Case 1. Each PE works under same voltage and frequency. There is no processing time and energy difference.
2. Case 2. $E_{process}$ can be tuned with *dynamic voltage scaling* (DVS). There are several different voltage levels for each PE to work on with.

In case 1, since S is a random variable, we want to find a configure of PE such that the energy is minimized with a guaranteed probability. For example, the distribution of random variable S is shown in Figure 1 (a). We first compute the cumulative distribution function (CDF) of S, which is shown in Figure 1 (b). “P2” represent the cumulative probability of each sampling rate value. Assume the processing rate R of each PE is 20, and the energy consumption E of each PE is 50. Then we need 4 PEs with 100% confidence while satisfying timing latency. The minimum total energy consumption is 200. But, this is the worst-case scenario. We can minimize energy while satisfying timing constraint with 90% confidence by using only 3 PEs. In this scenario, the total energy consumption is 150. We will discuss case 2 after introducing DVS.

2.3 Dynamic Voltage Scaling

DVS (Dynamic voltage scaling) is a technique that varies system's operating voltages and clock frequencies based on the computation load to provide desired performance with the minimum energy consumption. It has been demonstrated as one of the most effective low power system design techniques and has been supported by many modern microprocessors. Examples include Transmeta's Crusoe, AMD's K-6, Intel's XScale and Pentium III and IV, and some DSPs developed in Bell Labs [4].

Dynamic power, which is the dominant source of power dissipation in CMOS circuit, is proportional to $N \times C \times V_{dd}^2$, where N represent the number of computation cycles, C is the effective switched capacitance, and V_{dd} is the supply voltage [5,6,7]. Reducing the supply voltage can result in substantial power and energy saving. Roughly speaking, system's power dissipation is halved if we reduce V_{dd} by 30% without changing any other system parameters. However, this saving comes at the cost of reduced throughput, slower system clock frequency, or higher cycle period time (gate delay). The cycle period time T_c is proportional to $\frac{V_{dd}}{(V_{dd}-V_{th})^\alpha}$, where V_{th} is the threshold voltage and $\alpha \in (1.0, 2.0]$ is a technology dependent constant.

Let t represent the computation time and E represent energy, they are calculated as follows:

$$T_c = \frac{k \times V_{dd}}{(V_{dd} - V_{th})^\alpha} \tag{2}$$

$$t = N \times T_c = N \times \frac{k \times V_{dd}}{(V_{dd} - V_{th})^\alpha} \tag{3}$$

$$E = N \times C \times V_{dd}^2 \tag{4}$$

In Equation (2), k is a device related parameter. From Equations (3) and (4), we can see that the lower voltage will prolong the execution time of a node but reduces its energy consumption.

Low power design is of particular interest for the soft real time multimedia systems and we assume that each PE has multiple voltages available on the chip such that the system can switch from one level to another.

2.4 VAP Problem

For multi-voltage systems, assume there are maximum M different voltages in a voltage set $V = \langle V_1, V_2, \cdots, V_M \rangle$. An assignment is to assign a voltage to each PE that has been selected. Define an *assignment A* to be a function from domain U to range V, where U is PE set and V is the voltage set.

We define F to be the *cumulative distribution function* of the random variable S (abbreviated as *CDF*), where $F(t) = P(S < t)$. When S is a discrete random variable, the CDF $F(t)$ is the sum of all the probabilities associating with the computation times that are less than or equal to t. If S is a continuous random variable, then it has a *probability density function* (PDF). If assume the pdf is f, then $F(t) = \int_0^t f(s)ds$. Function F is nondecreasing, and $F(-\infty) = 0$, $F(\infty) = 1$.

We define the *voltage assignment with guaranteed probability (VAP)* problem as follows: Given M different voltage levels: V_1,V_2,$\cdots$,V_M, a sampling rate constraint S and a confidence probability P, find the proper number of PEs and an assignment of voltage level for each of the selected PEs to gives the minimum total energy consumption E with confidence probability P under sampling rate constraint S.

	V1	V2	V3
R	10	20	30
E	20	45	100

Fig. 2. The processing speed and energy consumption of each PE under different voltage levels

We will use dynamic programming to solve the VAP problem. Our algorithm *Volt_Ass* is shown in the next section. Here we give an example first. Figure 2 shows the processing rate R and energy consumption E of each PE under different voltage levels. The base energy E_{base} is 12. Suppose the confidence probability we need is 0.90. Then we check the table in Figure 1 (b), and find that the sampling rate S will not higher than a value, i.e., 60 samples/s. That is, $S \leq 60$. We denote the corresponding value to be L. Let N to be the number of PEs needed, we list the constraint equations to make it clear.

$$R_1 + R_2 + \cdots + R_N \geq L \tag{5}$$

We want to get

$$Min(E_1 + E_2 + \cdots + E_N) \tag{6}$$

After running our dynamic programming, we the number of PEs and the voltage assignment for the PEs. We need 3 PEs, the voltage level assignment of PEs is shown in Ass_1 of Table 1. The total energy consumption is $135 + 12 * 3 = 171$. For the case with no DVS, i.e., we only use the maximum processing speed, we will choose using only 2 PEs. The total energy is $200 + 12 * 2 = 224$. The assignment is shown in Ass_1 of Table 1. There is 23.7% difference. Hence, the solution is: use 3 PEs and let them all work under V_2, then the total energy consumption is 171, while satisfying the sampling rate with 0.90 confidence probability.

Table 1. The assignments with energy consumption 171 and 224 with sampling rate constraint 60

	Num.	**Speed**	**Volt.**	$E_{proc.}$	E_{base}
Ass_1	1	20	V_2	45	12
	2	20	V_2	45	12
	3	20	V_2	45	12
	Total	60		135	36
Total Energy = 171					
	N	**Speed**	**Volt.**	$E_{proc.}$	E_{base}
Ass_2	1	30	V_3	100	12
	2	30	V_3	100	12
	Total	60		200	24
Total Energy = 224					

3 The Algorithms

In this section, we will propose our algorithms to solve the energy-saving problem of sensor applications. The basic idea is to obtain the minimum total energy consumption by selecting proper number of PEs and doing voltage assignment. We consider two cases: Case 1: there is no voltage change for each PE. This case is simple, and we use *EOSP_Simple* to solve it. We will focus on case 2. Case 2: PE may work under different voltages. We proposed an algorithm, EOSP (*Energy-aware Online algorithm to satisfy Sampling rates with guaranteed Probability*). In this algorithm, we use *Volt_Ass* sub-algorithm to give the best PE number selection and voltage assignment for each selected PE.

3.1 The *EOSP_Simple* Algorithm for Case 1

The EOSP_Simple algorithm for case 1 is shown in Algorithm 3.2. In EOSP_Simple algorithm, we use the adaptive model [1] to solve energy saving problem for camera based sensors. The adaptive approach includes three steps: First, look ahead on performance parameters, such as image sampling rate and system

Algorithm 3.1. EOSP_Simple Algorithm for Case 1

Require: Several PEs, the sampling rate S, the guaranteed probability P.
Ensure: The number of PEs N with minimum total energy consumption E_{min} while satisfying S with P.

1: Look ahead on performance parameters: $S \leftarrow$ sampling rate.
2: Use fixed time window to collect data that will be processed.
3: Estimate the sample rate with different estimator, get the random variable S.
4: Build the cumulative distribution function (CDF) based on the distribution function of S.
5: $L \leftarrow$ the corresponding sampling rate value that has $Prob.(S \leq L) \geq P$ by looking CDF of S.
6: $N \leftarrow$ the number of PEs with $\sum_{i=1,\cdots,N} R_i \geq L$.
7: $E_{min} \leftarrow \sum_{i=1,\cdots,N} E_i$.
8: Output results: N and E_{min}.
9: Use online architectural adaptation to reduce energy consumption while satisfying timing constraints with guaranteed probability.

latency, by buffering input data coming in a given time window. Second, dynamic processing requirements prediction using a high efficient estimator activated at the end of every window period. Third, use an on-line architecture adaptation control policy. Since our design is for a non-stationary environments, the control policy varies with the environment but is stationary within a time window.

3.2 The EOSP Algorithm for Case 2

3.3 Volt_Ass Algorithm

To solve the voltage assignment problem, we use dynamic programming method. In our algorithm, table $D_{i,j}$ will be built. Each entry of table $D_{i,j}$ will store $E_{i,j}$, which is the minimum energy for i number of PEs with sampling rate j.

Lemma 1. *Given $E^1_{i,j}$ and $E^2_{i,j}$ with same sampling rate j, then $Min(E^1_{i,j}, E^2_{i,j})$ is selected to be kept.*

In every step of our algorithm, one more PE will be included for consideration. The (R, E) pair of each PE is stored in a local table B, which is shown in Figure 2. We first sort (R, E) pairs in B according to R in an ascending order, and represent them in the form: $(R_1, E_{R_1}), (R_2, E_{R_2}), \cdots, (R_M, E_{R_M})$. Hence, E_j is the energy consumption only for one PE with sampling rate j, The algorithm to compute $D_{i,j}$ are shown in Algorithm 3.3.

Theorem 1. *The energy consumption in $D_{I,L}$ obtained by algorithm* Volt_Ass *is the minimum total energy consumption with sampling rate L.*

Proof: By induction. **Basic Step:** When $i = 1$, there is only one PE and $D_{1,j} = B_j$. Thus, when $i = 1$, Theorem 1 is true. **Induction Step:** We need to show that for $i \geq 1$, if $D_{i,j}$ is the minimum total energy consumption of i PEs, then

Algorithm 3.2. The EOSP Algorithm

Require: M different voltages, Several PEs, the sampling rate S, the guaranteed probability P.
Ensure: The number of PEs N with minimum total energy consumption E_{min} while satisfying S with P.

Look ahead on performance parameters: $S \leftarrow$ sampling rate.
Use fixed time window to collect data that will be processed.
Estimate the sample rate with different estimator, get the random variable S.
Build the cumulative distribution function (CDF) based on the distribution function of S.
$L \leftarrow$ the corresponding sampling rate value that has $Prob.(S \leq L) \geq P$ by looking CDF of S.
Use algorithm *Volt_Ass* to obtain PEs assignment A with minimized E for the schedule graph.
$N \leftarrow$ the number of PEs with $\sum_{i=1,\cdots,N} R_i \geq L$.
$E_{min} \leftarrow \sum_{i=1,\cdots,N} E_i$.
Output results: N, A, and E_{min}.
Use online architectural adaptation to reduce energy consumption while satisfying timing constraints with guaranteed probability.

Algorithm 3.3. Volt_Ass algorithm to compute $D_{i,j}$

Require: Sampling rate L, M different voltage levels.
Ensure: $D_{i,L}$

Build a local table B according to M different voltage levels;
$R \leftarrow$ processing speed, $E \leftarrow$ energy;
Sort (R, E) pairs in B according to R in an ascending order, and represent them in the form: $(R_1, E_{R_1}), (R_2, E_{R_2}), \cdots, (R_M, E_{R_M})$;
$B_k \leftarrow E_k$ in each pair of B;
Start from the first PE, $D_{1,j} \leftarrow B_k$;
$I \leftarrow L/R_1$;
while $i \leq I$, **do**
    for all sampling rate j, **do**
        Compute the entry $D_{i,j}$ as follows:
        for all k in B_k, $j > k$ **do**
            $D_{i,j} = D_{i-1,j-k} + B_k$;
            Insert $D_{i,j-1}$ to $D_{i,j}$ and remove redundant energy value using Lemma 1;
        end for
    end for
end while
The energy in $D_{I,L}$ is the minimum total energy consumption with sampling rate L and the assignment can be obtained by tracing how to reach $D_{I,L}$;
Output $D_{I,L}$;

$D_{i+1,j}$ is the minimum total energy consumption of $i+1$ PEs. In the algorithm, since $j = k + (j - k)$ for each k in B_k, we try all the possibilities to obtain j. Then we use add the energy consumptions of two tables together. Finally, we using Lemma 1 to cancel the conflict energy values. The new energy consumption obtained in table $D_{i,j}$ is the energy consumption of i PEs at sampling rate k plus the energy consumption in $D_{i-1,j-k}$. Since we have used Lemma 1 to cancel redundant energy values, the energy consumption in $D_{I,L}$ is the minimum total energy consumption for sampling rate L. Thus, Theorem 1 is true.

From Theorem 1, we know $D_{I,L}$ records the minimum total energy consumption of the whole path within the timing constraint L. We can record the corresponding voltage assignment of each PE when computing the minimum system energy consumption in the algorithm *Volt_Ass*. Using these information, we can get an optimal assignment by tracing how to reach $D_{I,L}$.

It takes $O(M^2)$ to compute one value of $D_{i,j}$, where M is the maximum number of voltage levels. Thus, the complexity of the algorithm *Volt_Ass* is $O(S * M^2)$, where S is the given sampling rate constraint. Usually, S is upper bounded by a constant. In this case, *Volt_Ass* is polynomial.

4 Experiments

In this section, we conduct experiments with the EOSP algorithm on a set of benchmarks including 8-Stage Lattice filter,4-Stage Lattice filter, FFT1, Laplace, Karp10, Almu, Differential Equation Solver, RLS-Laguerre Lattice filter, Elliptic filter, and Voltera filter. The distribution of sampling rates is Gaussian. For each benchmark, we conduct a set of the experiments based on different configurations. K different voltages, $V_1, \cdots, V_K$, are used in the system, in which a voltage with level V_K is the quickest with the highest energy consumption and a voltage with level V_1 is the slowest with the lowest energy consumption.

We conduct the experiments using three methods. Method 1: Use the hard real-time and we need to guarantee to satisfy the varying sampling rates. Method

Table 2. The comparison of total energy consumption for Method 1, Method 2 and EOSP while satisfying sampling rate $S = 360$ for various benchmarks

4 voltages, S = 360										
Benchmarks	Num.	M1	M2	EOSP 0.8			M2	EOSP 0.9		
		E	E(0.8)	E(0.8)	% M1	% M2	E(0.9)	E(0.9)	% M1	% M2
8-stage Lattice	42	2655	2154	1762	33.6%	18.2%	1921	1467	44.8%	23.6%
4-stage Lattice	26	1576	1264	1023	35.1%	19.1%	1163	853	45.9%	26.7%
FFT1	28	1644	1312	1072	34.8%	18.3%	1228	937	43.0%	23.7%
Laplace	16	1032	852	689	32.6%	19.1%	736	531	48.1%	27.9%
Karp10	21	1305	1054	865	33.7%	17.9%	942	712	45.5%	24.4%
Almu	17	1053	827	682	35.2%	17.5%	762	583	44.6%	23.5%
Diff. Equ. Solver	11	767	624	508	33.7%	18.6%	567	425	44.6%	25.0%
RLS-laguerre	19	1255	1062	845	32.7%	20.4%	897	682	45.6%	24.0%
Elliptic	34	2384	1935	1589	33.4%	17.9%	1802	1345	43.6%	25.4%
Voltera	27	1668	1358	1108	33.6%	18.4%	1147	892	46.5%	22.2%
Average Reduction (%)					33.8%	18.5%	–	–	45.2%	24.6%

2: Use soft real-time and model the sampling rates as random variables. Method 3: Combine soft real-time and DVS together and use EOSP algorithm. We compare the results from EOSP algorithm with those from Method 1 and Method 2. The experiments are performed on a Dell PC with a P4 2.1 G processor and 512 MB memory running Red Hat Linux 9.0. In the experiments, the running time of EOSP on each benchmark is less than one minute.

The experimental results for Method 1 ("$M1$"), Method 2 ("$M2$"), and our EOSP algorithm with 4 voltages, are shown in Table 2. Column "Num." represents the number of nodes of each filter benchmark. Column "E" represents the minimum total energy consumption (μJ) obtained from the three different methods. Labels "E(0.8)" and "E(0.9)" represent the minimum total energy consumption when the guaranteed probability is 0.8 and 0.9, respectively. Column "% M1" and "% M2" under "EOSP" represents the percentage of reduction in total energy consumption, compared to Method 1 and Method 2, respectively. The average reduction is shown in the last row of the table.

The results show that our algorithm EOSP can significantly improve the performance of multi-voltage sensor nodes. For example, with 4 voltages, compared with Method 1, EOSP shows an average 33.8% reduction in total energy consumption while satisfying sampling rate constraint 360 with probability 0.80. The experimental results show that when the number voltages increases, the percentage of reduction on total energy increases correspondingly.

Through the experimental results from Table 2, we found that Method 1 doesn't explore the larger solution space for total energy consumption with soft real-time. Our EOSP algorithm combined both soft real-time and DVS, and can significantly reduce total energy consumption while satisfying sampling rates with guaranteed probability. It is efficient and provides overview of all possible variations of minimum energy consumptions comparing with the worst-case scenario generated by Method 1.

5 Related Work

Dynamic voltage and frequency scaling. Many researchers have studied on DVS [8,5,7,9,10,11]. Semeraro et al. designed a multiple clock domain (MCD) system to support fine-grained dynamic voltage scaling within a processor [12]. Yao et al. [13,14] and Ishihara et al. [6] studied the optimal schedule for DVS processors in the context of energy-efficient task scheduling. Both showed that it is most energy efficient to use the lowest frequency that allows an execution to finish before a given deadline. Ishihara et al. also showed that when only discrete frequencies are allowed, the best schedule is to alternate between at most two frequencies.

The International Technology Roadmap for Semiconductors [15] predicts that the future system will feature multiple supply voltages (V_{dd}) and multiple threshold voltages (V_{th}) on the same chip This enables the DVS, which varies the supply voltage according to workload at run time [13, 6]. The highest energy efficiency is achieved when voltage can be varied arbitrarily [16]. However, physical constraints of CMOS circuit limit the applicability of having voltage varying continuously. Instead, it is more practical to make multiple discrete voltages simultaneously available for the system [17].

Sensor networks. Sensor networks have wide applications in areas such as health care, military, environmental monitoring, infrastructure security, manufacturing automation, collecting information in disaster prone areas and surveillance applications [18]. In fact, the vision is that sensor networks will offer ubiquitous interfacing between the physical environment and centralized databases and computing facilities [19]. Efficient interfacing has to be provided over long periods of time and for a variety of environment conditions, like moving objects, temperature, weather, available energy resources and so on.

In many sensor networks, the DSP processor consumes a significant amount of power, memory, buffer size, and time in highly computation-intensive applications. However, sensor node can only be equipped with a limited power source ($\leq$ 0.5 Ah, 1.2 V) [20]. In some application scenarios, replenishment of power resources might be impossible. Therefore, power consumption has become a major hurdle in design of next generation portable, scalable, and sophisticated sensor networks. In computation-intensive applications, an efficient scheduling scheme can help reduce the power consumption while still satisfying the performance constraints. This paper focuses on reducing the total energy of sensor applications on architectures with multiple PEs and multiple voltage levels.

6 Conclusion

In this paper, we studied the proper PE number selection and voltage assignment problem that minimizes the total energy while satisfying sampling rates with guaranteed probability for sensor applications. We proposed a high efficient algorithm, EOSP (*Energy-aware Online algorithm to satisfy Sampling rates with guaranteed Probability*), in this paper. Our algorithm can give a much larger solution space of total energy minimization to be used for online adaptation control. A wide range of benchmarks has been tested on the experiments and the experimental results showed that our algorithm significantly improved the energy-saving of applications on computation-intensive sensor nodes.

References

1. Kallakuri, S., Doboli, A.: Energy conscious online architecture adaptation for varying latency constraints in sensor network applications. CODES+ISSS 2005, Jesey City, New Jersey, pp. 148–154 (September 2005)
2. Tongsima, S., Sha, E., Chantrapornchai, C., Surma, D., Passos, N.: Probabilistic Loop Scheduling for Applications with Uncertain Execution Time. IEEE Trans. on Computers 49, 65–80 (2000)
3. Weng, Y., Doboli, A.: Smart sensor architecture customized for image processing applications. IEEE Real-Time and Embedded Technology and Embedded Applications, 336–403 (2004)

4. Hua, S., Qu, G., Bhattacharyya, S.S.: Exploring the Probabilistic Design Space of Multimedia Systems. In: IEEE International Workshop on Rapid System Prototyping, pp. 233–240. IEEE Computer Society Press, Los Alamitos (2003)
5. Zhang, Y., Hu, X., Chen, D.Z.: Task Scheduling and Voltage Selection for Energy Minimization. DAC 40, 183–188 (2002)
6. Ishihara, T., Yasuura, H.: Voltage scheduling problem for dynamically variable voltage processor. In: ISLPED, pp. 197–202 (1998)
7. Shin, D., Kim, J., Lee, S.: Low-Energy Intra-Task Voltage Scheduling Using Static Timing Analysis. In: DAC, pp. 438–443 (2001)
8. Stan, M.R., Burleson, W.P.: Bus-Invert Coding for Low-Power I/O. IEEE Trans. on VLSI Syst. 3(1), 49–58 (1995)
9. Saputra, H., Kandemir, M., Vijaykrishnan, N., Irwin, M.J., Hu, J.S., Hsu, C.-H., Kremer, U.: Energy-conscious compilation based on voltage scaling. In: LCTES 2002 (June 2002)
10. Sakurai, T., Newton, A.R.: Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas. IEEE J. Solid-State Circuits SC-25(2), 584–589 (1990)
11. Chandrakasan, A., Sheng, S., Brodersen, R.: Low-Power CMOS Digital Design. IEEE Journal of Solid-State Circuits 27(4), 473–484 (1992)
12. Semeraro, G., Albonesi, D., Dropsho, S., Magklis, G., Dwarkadas, S., Scott, M.: Dynamic Frequency and Voltage Control for a Multiple Clock Domain Microarchitecture. In: 35th Intl. Symp. on Microarchitecture (November 2002)
13. Yao, F., Demers, A., Shenker, S.: A scheduling model for reduced cpu energy. In: 36th symposium on Foundations of Computer Science (FOCS), Milwankee, Wisconsin, pp. 374–382 (October 1995)
14. Li, M., Yao, F.: An efficient Algorithm for computing optimal discrete voltage schedules. SIAM J. Comput. 35(3), 658–671 (2005)
15. ITRS: International Technology Roadmap for Semiconductors. International SEMATECH, Austin, TX (2000), http://public.itrs.net/
16. Burd, T.B., Pering, T., Stratakos, A., Brodersen, R.: A dynamic voltage scaled microprocessor system. IEEE J. Solid-State Circuits 35(11), 1571–1580 (2000)
17. Intel: The Intel Xscale Microarchitecture. Technical Summary (2000)
18. Im, C., Kim, H., Ha, S.: Dynamic Voltage Scheduling Technique for Low-Power Multimedia Applications Using Buffers. In: Proc. of ISLPED (2001)
19. Rahimi, M., Pon, R., Kaiser, W., Sukhatme, G., Estrin, D., Srivastava, M.: Adaptive sampling for environmental robots. In: International Conference on Robotics and Automation (2004)
20. Berman, P., Calinescu, G., Shah, C., Zelikovsly, A.: Efficient energy management in sensor networks. Ad Hoc and Sensor Networks (2005)