ENERGY EFFICIENT SYSTEM PARTITIONING FOR DISTRIBUTED WIRELESS SENSOR NETWORKS

Alice Wang and Anantha Chandrakasan
Massachusetts Institute of Technology
{aliwang, anantha} @mtl.mit.edu

ABSTRACT

In this paper a scheme for efficient system partitioning of computation in wireless sensor networks is presented. Local computation of the sensor data in wireless networks can be highly energy-efficient, because redundant communication costs can be reduced. It is important to develop energy-efficient signal processing algorithms to be run at the sensor nodes. This paper presents a technique to optimize system energy by parallelizing computation through the network and by exploiting underlying hooks for power management. By parallelizing computation, the voltage supply level and clock frequency of the nodes can be lowered, which reduces energy dissipation. A 60% energy reduction for a sensor application of source localization is demonstrated. The results are generalized for finding optimal voltage and frequency operating points that lead to minimum system energy dissipation.

1. INTRODUCTION

In a variety of military and civil applications, large arrays of macrosensors are being used for environment sensing and monitoring. Currently there is a shift towards networks of microsensor nodes, for reasons such as lower cost and ease of deployment. Research has shown that doing signal processing locally at the sensor node level can be highly energy-efficient because this reduces redundant data transmission in the network [1]. This brings new challenges in the design of energy-efficient signal processors and sensor algorithm implementation for the microsensor nodes, which typically are energy-constrained.

One well established technique at the chip level is to exploit parallelism and voltage scaling [2]. By parallelizing computation, the clock rate can be reduced allowing for a reduction of the supply voltage [3]. This paper extends this notion for wireless systems by proposing methods for system partitioning of computation in wireless sensor nodes.

2. WIRELESS SENSOR NETWORKS

A wireless microsensor node is typically energy-constrained, so therefore it is important to prolong the lifetimes of the sensors by using energy-efficient design. One way to improve energy-efficiency, is to have sensor collaboration between the nodes through the wireless network. Closely located sensors have highly correlated data, so to reduce redundant information in the network, sensors are grouped in clusters and signal processing is done locally within a cluster. Through signal processing, the nodes can extract the important and relevant information, therefore reducing communication costs. Therefore, it is important to design low-power signal processors for the node, and also to consider energy-efficient system partitioning of the computation among the sensor nodes.

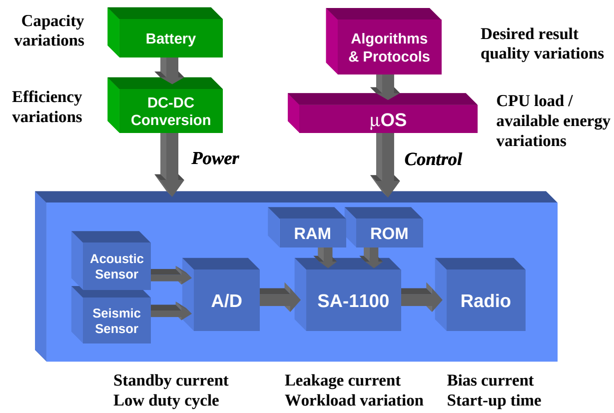

Figure 1: Architectural overview of a sensor node.

For this analysis, we will assume that the microsensors within a cluster are homogenous. We will assume that each sensor node has a battery, low power radio, microphone, A/D, and is equipped with a low power StrongARM (SA-1100) microprocessor for computation. Fig. 1 shows the architectural overview of a sensor node. Digitized data from analog sensors are sent to an SA-1100 processor, which communicates with adjacent nodes through a 2.4 GHz radio transceiver. Using a DC-DC converter, the voltage supply and clock frequency of the SA-1100 can be dynamically changed as the system adapts to changing conditions. The SA-1100 can be programmed to run at clock speeds of 74-206 MHz with voltage supplies ranging from 0.85-1.44 V.

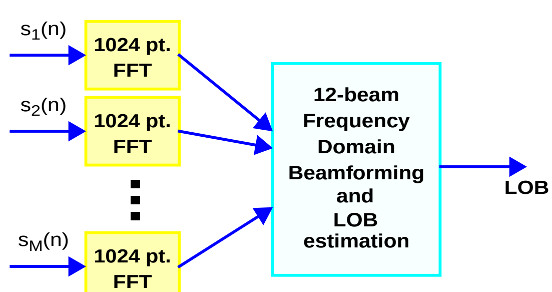

Figure 2: Block diagram of Line of Bearing (LOB) estimation application.

An application for distributed wireless sensors is source localization by Line of Bearing (LOB) estimation of acoustic sources. LOB estimation for source localization is important in many military and civil surveillance systems. Fig. 2 shows a block diagram of the LOB estimation algorithm. The first part of LOB estimation is to transform the sensor data into the frequency domain through a 1024 pt. FFT. Then the FFT coefficients are combined using a frequency-domain delay-and-sum beamforming algorithm. They are beamformed in 12 uniform directions. The direction which has the most signal energy is the LOB of the source. Multiple LOB's can be combined at the basestation, to calculate the source's location.

3. SYSTEM PARTITIONING

For a fixed latency requirement, energy dissipation can be reduced by distributing computation among the sensor nodes. By parallelizing the computation, the voltage supply and clock speed can both be lowered, in order to reduce energy dissipated. The energy dissipated by the StrongARM is modeled by

$$E_{\text{comp}} = NCV_{dd}^2 \qquad (1)$$

where N is the number of clock cycles, C is the total average capacitance being switched by the executing program, per clock cycle, and V_{dd} is the operating voltage [4]. The relation between the clock speed, f, and the voltage supply can be modeled as

$$f \le \frac{K(V_{dd} - V_T)^{\alpha}}{V_{dd}} \approx K(V_{dd} - c) \qquad (2)$$

where α, K, c and V_T, are processor dependent variables. The frequency-voltage relation is linearized in order to simplify the calculations. The constant , c, is only necessary for short-channel effects or when V_{dd} is close to the threshold voltage, V_T of the devices.

For example, if computation, C, can be computed using two parallel functional units instead of one, then the throughput is increased by two. However if the latency is fixed, instead by using a clock frequency of $f/2$, and voltage supply of $V_{dd}/2$, then the energy is reduced by 4 times over the non-parallel case.

Fig. 3 demonstrates how energy dissipation is reduced by partitioning the computation for LOB estimation in a 7 sensor cluster. The partitioning can be done in 2 ways.

In system partition #1 all sensors (S1-S7) sense data, and transmit the raw data to the cluster-head, where the seven FFT's and beamforming is run. This technique will be called the *direct* technique. The cluster-head is a sensor within the cluster, randomly chosen to do the beamforming and LOB estimation task and to transmit the result back to the end-user. Note that due to the assumption that all sensors are homogeneous, the cluster-head is also energy-constrained. In order to be within the end-user's latency requirement of 20 msec, all of the computation is run at the cluster-head at the fastest clock speed, f=206 MHz and at a voltage supply of 1.44V. The energy dissipated by the computation is 6.2mJ and the latency is 19.2 msec.

In system partition #2, the FFT task is parallelized. This will be called the *distributed* technique. In this scheme, the sensor nodes sense data and perform the 1024 pt. FFT's on the data before transmitting the FFT data to the cluster-head. At the cluster-head, the beamforming and LOB estimation is done. Without dynamic voltage scaling, performing the FFT's with the distributed technique has no energy advantage over the direct technique. This is because performing the FFT's at the sensor node does not reduce the amount of data that needs to be transmitted. Thus the communication costs remain the same.

However, by adding circuitry to perform dynamic voltage scaling (DVS), the node can take advantage of the parallelized computation load by allowing voltage and frequency to be scaled while still meeting latency constraints of 20 msec. For example, if the FFT's at the sensor nodes are run at 0.85V voltage supply and 74 MHz clock speed while the beamforming algorithm is run at 1.17V voltage supply and 162 MHz clock speed, then with latency of 18.4 msec we only dissipate 3.4mJ of energy, which is a 45.2% improvement in energy dissipation. This example shows that efficient system partitioning by parallelism can yield large energy reductions.

In order to determine the acceptable latency, we need to examine the timing diagrams associated with the algorithm. Fig. 4a

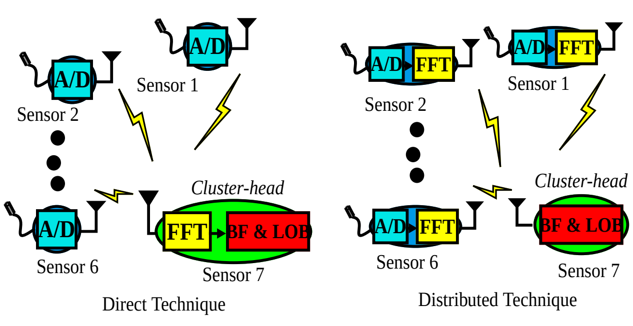

Figure 3: a) System partition #1: Direct Technique: All of the computation is done at the cluster-head. b) System partition #2: Distributed Technique: Distribute the FFT computation among all sensors.

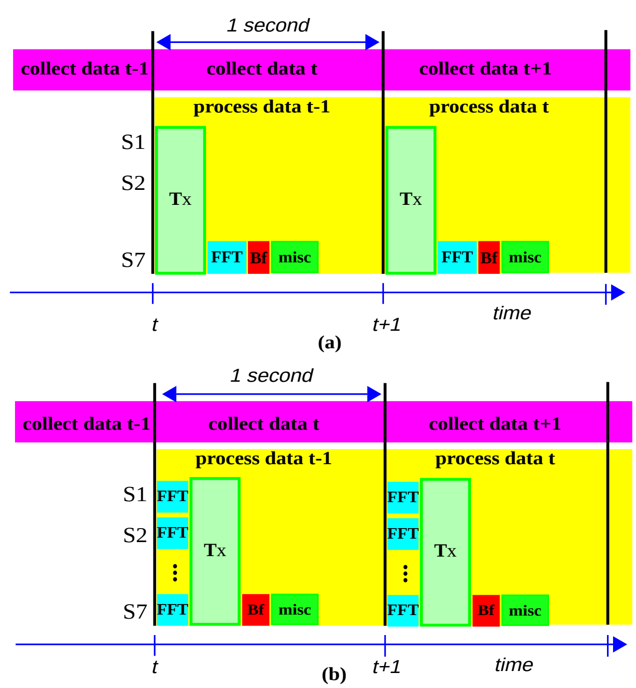

Figure 4: (a) Timing diagram for system partition #1 (b) Timing diagram for system partition #2.

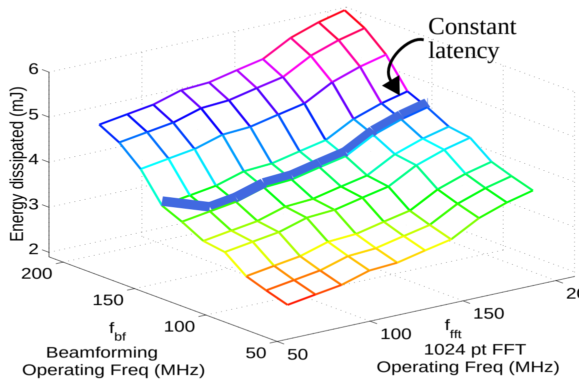

Figure 5: Minimum energy dissipated for system partition #1, on the StrongARM for LOB estimation of 7 sensors as a function of all possible operating frequencies for the FFT and the beamforming, $<f_{fft}, f_{bf}>$.

shows the timing diagram for system partition #1, and Fig. 4b shows the timing diagram for system partition #2. For both scenarios, at time t, the sensors begin sensing acoustic data from the microphones. We are assuming a 1 kHz sampling frequency, so that after 1 second, 1024 samples are buffered, and the nodes can begin processing the data. Note, that while the computation is being done, at the same time, new data is being collected from the microphones. From these diagrams it is easy to see how the computation is parallelized to the nodes. In Fig. 4a, you can see the nodes transmitting their data to the cluster-head, where the LOB estimation is performed. In Fig. 4b, the FFT is processed at the nodes, before being transmitted to the cluster-head.

The constraint of the allowable latency of the computation is $T_{comp} \le T_{req} - \tau_{tx} - \tau_{misc}$, where T_{req} is the required latency of the end-user, τ_{tx} is the time allotted for the nodes to transmit to the cluster-head, and τ_{misc} incorporates other latencies due to miscellaneous tasks. We have assumed that T_{req} is 100 msec, τ_{tx} is 40 msec, τ_{misc} is 20 msec, so that $T_{comp} \le 20$ msec.

In order to use the proposed system partitioning technique, T_{comp} must be within the range of:

$$\frac{N_{fft} + N_{bf}}{f_{max}} \le T_{comp} \le \frac{N_{fft} + N_{bf}}{f_{min}} \qquad (3)$$

where N_{fft} and N_{bf} are the cycle counts for the two tasks, and f_{max} and f_{min}, are the maximum and minimum frequencies possible. If T_{comp} violates the lower limit, then there is never enough time to complete the computation. If the upper limit is violated, then the computation will always be operated at the lowest frequency and voltage levels for minimal energy dissipation.

4. OPTIMAL VOLTAGE-FREQUENCY SCHEDULING

In this paper a method is suggested for finding the optimal operating voltage and frequency for a distributed sensor system. This is important, because the system should adjust operating voltages and frequencies of the sensor nodes to changes in system parameters (e.g. number of sensors, number of samples, etc.). Fig. 5 shows a plot of the energy dissipated for system partition #1 for all possible frequency operating points $<f_{fft}, f_{bf}>$ for a 7 sensor cluster based on StrongARM SA-1100 measurements. The solid line denotes the curve for total constant latency of 20 msec. Typically, the equal latency curve is highly non-linear. And as the range of voltages and frequencies widens, it becomes difficult to find the optimal operating point.

In order to find the optimal operating voltage, we want to minimize the total energy for an M sensor cluster

$$E_{tot} = MN_{fft}CV_{fft}^2 + N_{bf}CV_{bf}^2 \qquad (4)$$

with the latency constraint that

$$T_{comp} \ge \tau_{fft} + \tau_{bf} = \frac{N_{fft}}{f_{fft}} + \frac{N_{bf}}{f_{bf}} \qquad (5)$$

V_{fft} and V_{bf} are the operating voltages of the two tasks. From Eq. 4 in order to minimize energy, voltage and frequency should be minimized, but from Eq. 5 the frequency must be large enough to satisfy the latency constraint.

To find the optimal voltage and frequency operating points, Eq. 2 is substituted into Eq. 5, and a Lagrangian minimization problem is solved to get the relation between V_{bf} and V_{fft}.

$$(V_{bf} + c) = \sqrt[3]{M}(V_{fft} + c) \qquad (6)$$

Eq. 6 is substituted this back into Eq. 5 and solve for V_{fft}, V_{bf}, f_{fft}, and f_{bf}.

$$V_{fft} \ge \frac{1}{T_{comp}K}\left[N_{fft} + \frac{N_{bf}}{\sqrt[3]{M}}\right] + c \qquad (7)$$

$$V_{bf} \ge \frac{\sqrt[3]{M}}{T_{comp}K}\left[N_{fft} + \frac{N_{bf}}{\sqrt[3]{M}}\right] + c \qquad (8)$$

$$f_{fft} \le \frac{(\sqrt[3]{M}N_{fft} + N_{bf})}{\sqrt[3]{M}T_{comp}} \qquad (9)$$

$$f_{bf} \le \frac{\left(\sqrt[3]{M}N_{fft} + N_{bf}\right)}{T_{comp}} \qquad (10)$$

These equations for the frequency and voltage levels indicate that in general it is desirable to run the parallelized task (FFT) at lower voltage and lower frequency than that of the non-parallelized task (beamforming). Note, that the StrongARM does not have continuous voltage and frequency levels, which are assumed by the analysis. A practical quantization scheme is as follows. First calculate f_{bf} from Eq. 10, and round up to the next closest frequency point. Then use Eq. 5, to find f_{fft}. These frequencies map directly into the minimum possible voltage supply levels $<V_{fft}, V_{bf}>$. On average this scheme, will lead to predictions which are the same as those operating points which give the minimum energy dissipated.

Fig. 6 compares the energy dissipated for the direct technique vs. that for the distributed technique with optimal voltage scheduling as M is increased from 3-10. This plot shows that a 30-65% energy reduction can be achieved with the system partitioning scheme. In order to calculate N_{fft} and N_{bf}, as a function of M sensors, the FFT and beamforming algorithms were run on the SA-1100:

$$N_{fft} = 200.73 \text{ kcycles} \qquad (11)$$

$$N_{bf} = 319.3M + 341.6 \text{ kcycles} \qquad (12)$$

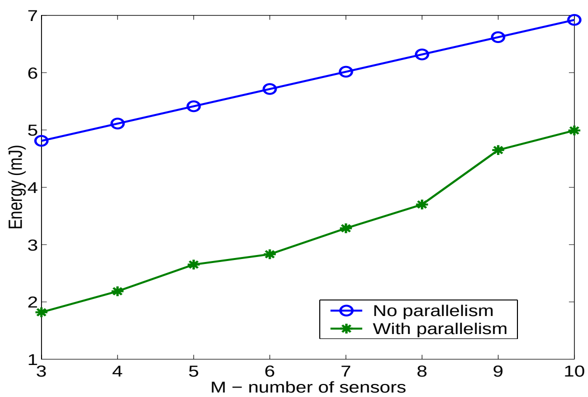

Figure 6: Comparing energy dissipated for the direct techinque vs. the distributed technique.

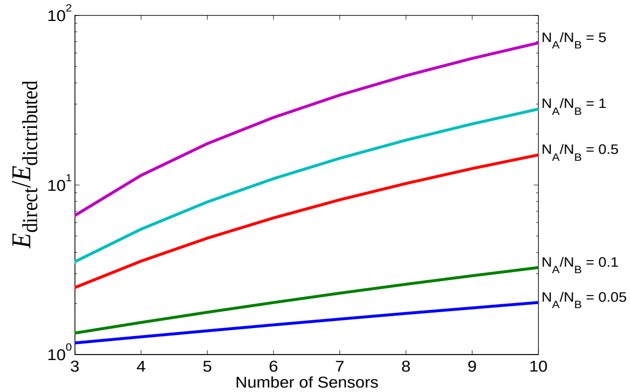

Figure 7: The ratio of energy dissipated for the direct technique vs. energy for the distributed technique as a function of the cycle ratio and the number of sensors.

Measurements taken from the SA-1100 show that the processor dependent variables, K=239.28 MHz/V and c = 0.5V.

5. GENERALIZED RESULTS

This partitioning scheme can be generalized to any sensor application where parallelism can be exploited. If there are two tasks, A and B, each of which can be characterized by their cycle counts, N_A and N_B, respectively. A is the task to be parallelized to the M sensor nodes, and B is non-parallelized. Now the two system partitioning schemes can be compared, the first being the serial scheme where there is no parallelization and the second is the optimal scheme where task A is parallelized and each task is run at the optimal voltage-frequency. Also the bandwidth between the nodes and cluster-head does not change for the two schemes. In the serial scheme, the frequency of the cluster-head is set to

$$f_{CH} = \frac{MN_A + N_B}{T_{\text{comp}}} \qquad (13)$$

The ratio of E_{direct}, the energy of the direct technique to $E_{\text{distributed}}$, the energy of the optimal distributed technique, is calculated to be approximately

$$\frac{E_{\text{direct}}}{E_{\text{distributed}}} = \frac{(M \cdot N_A/N_B + 1)^3 + D(M \cdot N_A/N_B + 1)^2}{M\left(N_A/N_B + M^{-1/3}\right)^3 + D \cdot M\left(N_A/N_B + M^{-1/3}\right)\left(N_A/N_B + M^{-2/3}\right)} \qquad (14)$$

where $D=2cKT/N_B$. Fig. 7 shows a plot of the energy ratio, $E_{\text{direct}}/E_{\text{distributed}}$, as M increases and for differing cycle ratios, N_A/N_B. In general as the number nodes increases, there is more opportunity for parallelization, and there is an increase in energy savings. For a particular M, when the computation for the parallelized task A, is relatively large (e.g., N_A/N_B=5), then there is a more energy savings. This means that a large part of the computation is being parallelized, and therefore the voltage and frequency can be reduced a great deal. We can achieve up to 70x energy reduction over the serial scheme. However, even when the computation in task A, is small compared to that for task B (e.g. N_A/N_B=0.05), then only about 2x energy savings can be achieved. These results assume a large range of voltage-frequency operating points.

If we take the limit of Eq. 14 as N_A goes to infinity, or as the amount of computation for task A gets larger and larger, then the upper limit of the energy savings is M^2. This shows that by using parallelism, there is a potential for a great deal of energy savings.

6. CONCLUSIONS

Partitioning the computation for LOB estimation in wireless sensors is important for energy-efficiency. By parallelizing computation, energy reductions of up to 60% can be achieved in a source localization application. Finding the optimal voltage and frequency operating points becomes difficult as the number of voltage levels and frequency levels increases. A method for finding the optimal voltage and frequency levels is introduced, which can be computed as a function of the number of sensors in the cluster. Using measurements from the StrongARM SA-1100 this technique is verified for a source tracking algorithm. Results for the general case are shown.

REFERENCES

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” *Proc. HICSS ‘00*, January 2000.

[2] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. Srivastava, “Power Optimization of Variable Voltage Core-Based Systems”, *Proc. 1998 of the Design Automation Conference*, June 1998.

[3] V. Gutnik and A. Chandrakasan, “Embedded Power Supply for Low-Power DSP”, *IEEE Transactions on VLSI Systems*, Vol. 12, pp. 425-435, December 1997.

[4] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.

ACKNOWLEDGMENTS

The authors would like to thank M. Bhardwaj and A. Sinha for their valuable discussions. We also would like to acknowledge N. Srour, T. Pham, K. Tran and A. Nguyen from the Acoustic Signal Processing Branch at ARL for providing us with the algorithms for LOB estimation. A. Wang is supported by the Lucent Fellowship.

This research is sponsored by the Defense Advanced Research Project Agency (DARPA) Power Aware Computing/Communication Program and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0551.