Resource and Energy-aware Distributed Block-based LU Decomposition on Wireless Sensor Networks

Sherine Abdelhak, Jared Tessier, Soumik Ghosh, Magdy Bayoumi
The Center for Advanced Computer Studies, University of Louisiana at Lafayette
{spa9242, jst2777, sxg5317, mab} @ cacs.louisiana.edu

Introduction

Wireless sensor networks are starting to mature into the next generation of nodes and networks, where applications are beginning to realistically consider scenarios where wireless sensor networks can be used for distributed signal processing, breaking away from the current generation of microcontroller applications.

Signal conditioning and processing in sensor networks refers to activities such as amplification, digitizing, filtering and other signal processing tasks. Signal processing in sensor networks have been reported for diverse categories of applications including distributed 2D FIR and IIR filtering, object tracking or temporal event tracking for a given set of sensors, classification, and decentralized Kalman Filtering. These tasks are, however, computationally intensive and strain the energy resources of any single computational node in a wireless sensor network (WSN). Moreover, most sensor nodes do not have the computational and memory resources to complete many of these signal processing tasks repeatedly.

LU-decomposition constitutes a *kernel* for linear algebra and signal processing. It is at the heart of many signal processing applications [2], particularly distributed signal processing applications. In this paper, we propose a general scheme for distributing the LU decomposition of an $N \times N$ matrix. Our algorithm is based on block-based LU decomposition. It achieves a balanced tradeoff between parallelism and communication cost. The job allocation is resource and energy-aware. Fault tolerance through redundancy was also achieved. Moreover, we propose an efficient scheduling and mostly decentralized synchronous TDMA MAC protocol to accomplish the computation with minimum latency. Exhaustive analysis and experiments were done to estimate the latency and energy for different matrix sizes with different partition sizes. This work lays the foundation for implementing other numerical methods for distributed signal processing over resource constrained wireless sensor nodes.

Proposed Algorithm

Given the limited memory size of the sensor nodes, large matrices cannot be processed or even stored on a single node. As such, block LU decomposition should be performed in which an $N \times N$ matrix A can be partitioned into at least 4 matrices [2] (refer to Eq.1).

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = \left[\begin{array}{c|c} L_{11} & 0 \\ \hline L_{21} & L_{22} \end{array}\right] \cdot \left[\begin{array}{c|c} U_{11} & U_{12} \\ \hline 0 & U_{22} \end{array}\right] \quad (1)$$

The sizes of the matrices depend on the chosen partition size *'b'*. A_{11} is a $b \times b$ matrix on which LU decomposition operation (using Crout's algorithm) is applied. A_{12} is a $b \times (n-b)$ matrix on which upper matrix decomposition is applied. A_{21} is an $(n-b) \times b$ matrix on which lower matrix decomposition is applied, and A_{22} is an $(n-b) \times (n-b)$ matrix. A new matrix A_{new} is formed such that ($A_{new} = A_{22} - L_{21}U_{12}$). If A_{new} is big enough to be partitioned again, the same block-based procedure is applied; otherwise, Crout's algorithm is used to find its LU decomposition.

We identified the independent operations involved in the block-based LU at each iteration. Each iteration is called a round. The operations in each iteration can be done in 3 stages. **Stage1**: Apply Crout's algorithm for finding the LU decomposition on A_{11} to obtain L_{11} and U_{11}. Compute the 1st row of A_{12} and 1st column of A_{21}. **Stage2**: Compute A_{12} completely using L_{11} from Stage1 to obtain U_{12}, similarly process A_{21} completely using U_{11} from Stage1 to obtain L_{21}. Compute the 1st term of each column of $L_{21}U_{12}$ for A_{new}. **Stage3**: Compute A_{new} using the results from Stage2. Note that the number of rounds depends on *N* (the matrix size) and *b* (the block size) and is equal to *[N/b]*.

We distinguish 4 kinds of sensor nodes involved in the computation, in terms of the type of operations assigned to them: **group1** (grp1) with odd-numbered IDs is responsible for computing the upper matrix decomposition (we call this operation β), **Node** with ID2 (N_2) is responsible for computing the LU decomposition using Crout's algorithm (LU operation), **group2** (grp2) with even-numbered IDs is responsible for computing the upper matrix decomposition (α operation), and **ClusterHead** (CH) is responsible for the partitioning and regrouping of the original matrix. Note that no special hardware or resource requirements are assumed for the ClusterHead. General formulae were developed for the computation requirements, number of messages sent, number of messages received and the storage requirements for each node- in terms of the number of rounds, and the ID of each node. The number of nodes used is ($2.[N/b]^2$-1). An example of a 13×13 matrix is shown below, each matrix partition is given a name for the sake of simplicity. Figure 2 shows the detailed operations taken by each node and all the requirements needed for *b=3*. For this example, the number of rounds is 4, and the number of nodes required is 7. The total number of messages transmitted is 35.

$$\mathbf{A=}\left[\begin{array}{c|c|c|c} w_{3x3} & a_{3x3} & b_{3x3} & c_{3x4} \\ \hline x_{3x3} & l_{3x3} & m_{3x3} & n_{3x4} \\ \hline y_{3x3} & o_{3x3} & q_{3x3} & r_{3x4} \\ \hline z_{4x3} & p_{4x3} & s_{4x3} & t_{4x4} \end{array}\right]$$

Figure 1: A 13x13 matrix partitioned into 3x3 blocks.

In the proposed algorithm, the communication frame consists of three main stages. **Querying stage**: during which the CH queries the nodes within its cluster for their voltage levels and their available memory. An energy-

aware and memory-aware allocation is done by the CH to select the nodes belonging to each group. Moreover, a *CandidateList* is formed for each active node- containing *CandidateNodes* that have *approximately* same resources. This is later used for redundancy and fault tolerance, which will take place during the *JobDelegation* period (if necessary).

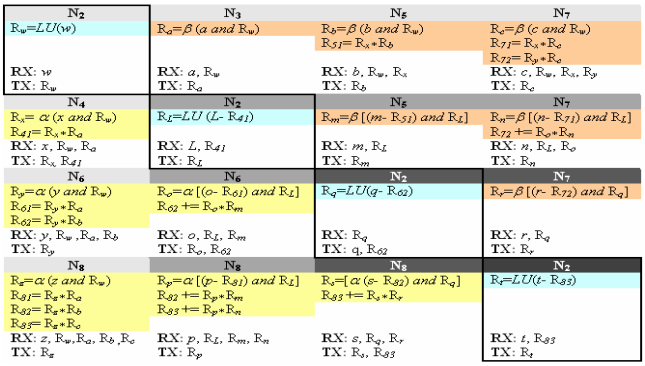

Figure 2: Detailed analysis for a 13x13 matrix and block size 3.

Computing Stage: during which the CH unicasts the matrix partitions to the nodes participating in the computation, and collects the results. **Final Stage**: the CH regroups the result. It is noteworthy that the time slot allocation is done in a mostly-decentralized way, since the nodes- based on their IDs- know which time slot to occupy, with minimum interference from the CH.

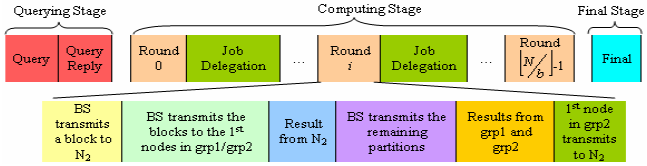

Figure 3: Proposed frame format.

Experimental Results

We setup a testbed of Telos Rev. B motes to test distributed computing applications. The Telosb is powered by two AA batteries and features a Chipcon 2420 radio; a 8MHz Texas Instruments MSP430 microcontroller and an integrated on board antenna with 50m range indoors and 125m range outdoors among other peripherals and sensors. The power consumption of a system under test could be determined in two ways: using a clamp-on current probe or a shunt resistor [3]. We used the latter after making sure that it is the most accurate one. Figure 4 shows the testbed setup for the power measurement. A laptop is configured to be able to send and receive control and data packets from the motes through the use of a java program as shown in Figure 6. The java program is a multi-threaded socket-based program that communicates with a serial forwarding program.

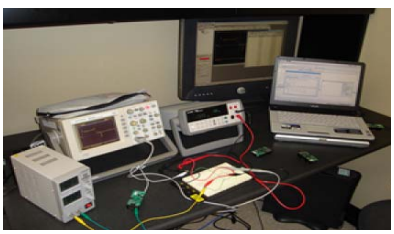

Figure 4: Setup for the power measurement. The USB attached mote acts as the ClusterHead. Motes are distributed around the lab.

The average power was approximately 40.6mW. The maximum power ranged between 40.9 and 41.2mW. The lowest energy, however, corresponds to a 3x3 block size and was approximately 0.51mJ.

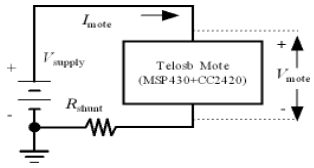

Figure 5: Circuit of shunt-resistor-based power measurement

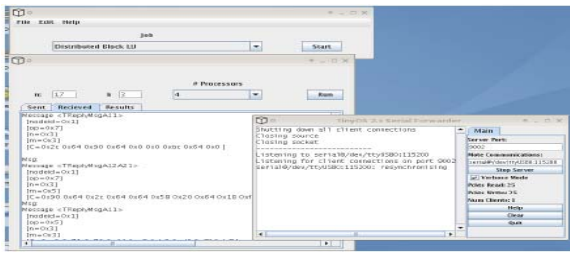

Figure 6: Java user interface for control of jobs and visualizing results from the operations.

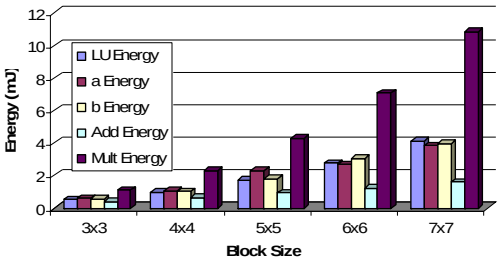

Figure 7: Total energy per operation (mJ) vs. the block size.

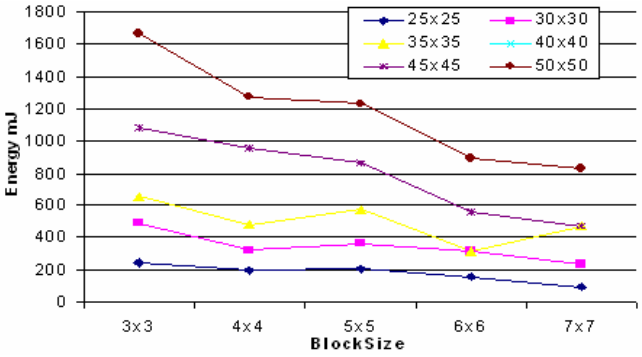

Figure 8: Total energy (mJ) for different matrix sizes and different partition sizes.

References

[1] C. Nikias, A. Chrysafis, and A. Venetsanopoulos, *The LU decomposition theorem and its implications to the realization of two dimensional digital filters*, IEEE transactions on acoustics, speech, and signal processing, vol. 33, no. 3, pp. 694–711, 1985.

[2] V. Daga, G. Govindu, V. Prasanna, S. Gangadharapalli, and V. Sridhar, *Efficient Floating-point based Block LU Decomposition on FPGAs*, International Conference on Engineering of Reconfigurable Systems and Algorithms, pp. 21–24.

[3] A. Milenkovic, M. Milenkovic, E. Jovanov, D. Hite, and D. Raskovic, *An environment for runtime power monitoring of wireless sensor network platforms*, System Theory, 2005. SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on, pp. 406–410, 20-22 March 2005