Performance of the Wavelet Decomposition on Massively Parallel Architectures

Tarek A. El-Ghazawi

Department of Electrical and Computer Engineering

The George Washington University

Washington, D.C. 20052

Tel: (202) 994-5905; Fax: (202) 994-0227

tarek@seas.gwu.edu

and

Jacqueline Le Moigne

Applied Information Science Branch, Code 935

NASA Goddard Space Flight Center

Greenbelt, MD 20771

Tel: (301) 286-8723; Fax: (301) 286-1776

lemoigne@backserv.gsfc.nasa.gov

Abstract

Traditionally, Fourier Transforms have been utilized for performing signal analysis and representation. But although it is straightforward to reconstruct a signal from its Fourier transform, no local description of the signal is included in its Fourier representation. To alleviate this problem, Windowed Fourier transforms and then Wavelet transforms have been introduced, and it has been proven that wavelets give a better localization than traditional Fourier transforms, as well as a better division of the time- or space-frequency plane than Windowed Fourier transforms. Because of these properties and after the development of several fast algorithms for computing the wavelet representation of any signal, in particular the Multi-Resolution Analysis (MRA) developed by Mallat, wavelet transforms have increasingly been applied to signal analysis problems, especially real-life problems, in which speed is critical. In this paper we present and compare efficient wavelet decomposition algorithms on different parallel architectures. We report and analyze experimental measurements, using NASA remotely sensed images. Results show that our algorithms achieve significant performance gains on current high-performance parallel systems, and meet scientific applications and multimedia requirements. The extensive performance measurements collected over a number of high-performance computer systems have revealed important architectural characteristics of these systems, in relation to the processing demands of the wavelet decomposition of digital images.

Keywords: Parallel Processing, Image Processing, Wavelets, Experimental Performance Evaluations.

1. Introduction

Traditionally, Fourier transforms have been utilized for signal analysis and reconstruction. But although it is straightforward to reconstruct a signal from its Fourier transform, no local description of the signal is included in its Fourier representation, as shown in equation (1):

$$F(x)(f) = \int x(t)\, e^{-ift}\, dt \qquad (1)$$

To alleviate this problem, Windowed Fourier Transforms, and as a special case Gabor Transforms [1], have been introduced. The signal is analyzed after filtering by a fixed window function, so these transforms have the localization property that traditional Fourier transforms do not have. See equation (2) where a window function g(t) is used:

$$WF(x)(f,\tau) = \int x(t)\, g(t-\tau)\, e^{-ift}\, dt \qquad (2)$$

However, since the envelope of the signal is the same for all frequencies, a windowed Fourier transform uniformly samples the time- or space-frequency plane. Depending on the application, for example speech analysis or image feature extraction, it can be of interest to have a more flexible division of the time- or space-frequency plane to provide more "time- or space-details" at high frequencies. Wavelet transforms provide this type of sampling by filtering the signal with the translations and dilations of a basic function, called the "mother wavelet", equation (3).

$$Wav(x)(a,b) = |a|^{-1/2} \int x(t)\, \psi\left(\frac{t-b}{a}\right) dt \qquad (3)$$

where $\psi(t)$ is the "Mother Wavelet," and a and b are the scale and translation variables, respectively.

In the image processing domain, wavelet transforms have been proven to be very useful for such tasks as image compression and reconstruction, feature extraction, and image registration [1-6]. Furthermore, fast algorithms and particularly the multi-resolution scheme developed by Mallat [4,7,8] have increased the importance of wavelets for on-line processing of imagery data. The speed of such processing is especially important for managing remotely sensed data whose already massive amounts is growing even bigger with such programs as NASA's Earth Observing System (EOS).

In this study, we are investigating the parallel implementation and performance of the Mallat MRA algorithm on parallel architectures. Coarse-grain algorithm mappings for the Intel Paragon, the Cray T3D, the HP/Convex SPP-1000, and the Beowulf/Hrothgar network of PC's are developed. Extensive measurements are collected, analyzed and compared with the fine-grain MasPar experimental results [9-11]. Test image data from NASA's Landsat-Thematic Mapper (TM) and various filter sizes were used. The results will show that the parallel algorithms can achieve orders of magnitude performance improvement on contemporary high-performance computing systems, when compared to typical desktop workstations. Such performance can

satisfy real-time image processing needed for large scientific databases, such as the NASA's Earth Science Data and Information System (ESDIS) project and all multimedia applications. This paper is organized as follows. Section 2 provides an overview of the discrete wavelet transform and the Mallat algorithm. Section 3 provides an overview to the massively parallel architectures that were used in this study, which includes MasPar, Cray T3D, Intel Paragon, HP/Convex SPP-1000, and the Hrothgar/Beowulf network of PC's. Section 4 discusses the algorithms and the implementation issues on different high-performance computing architectures. Scalability and timing results are presented and discussed in section 5. Conclusions are given in section 6.

2. Multi-Resolution Wavelet Decomposition

As described in section 1, a wavelet transform is defined by the translations and the dilations of a basic function called the "Mother Wavelet." Depending on the application, continuous or discrete transforms may be utilized. Special conditions are imposed on Mother Wavelets that lead to orthonormal bases of wavelets, which are particularly useful for data reconstruction [3]. In this paper, we will only consider wavelet transforms for the processing and analysis of 2-D image data. Thus, discussion will focus on discrete wavelets, and particularly those forming orthonormal bases.

According to Mallat [4], an orthonormal basis of wavelets can be defined by a scaling function and its corresponding conjugate filter L. In this case, the wavelet decomposition of an image is similar to a quadrature mirror filter decomposition with the low-pass filter L and its mirror high-pass filter H. This decomposition of a 2-D image, also called "Multi-Resolution Analysis" (MRA) assumes that the multi-resolution representation of the image space is "separable." This means that the two axes x and y can be treated independently in the decomposition as well as in the reconstruction. This decomposition is summarized in Figure 1.

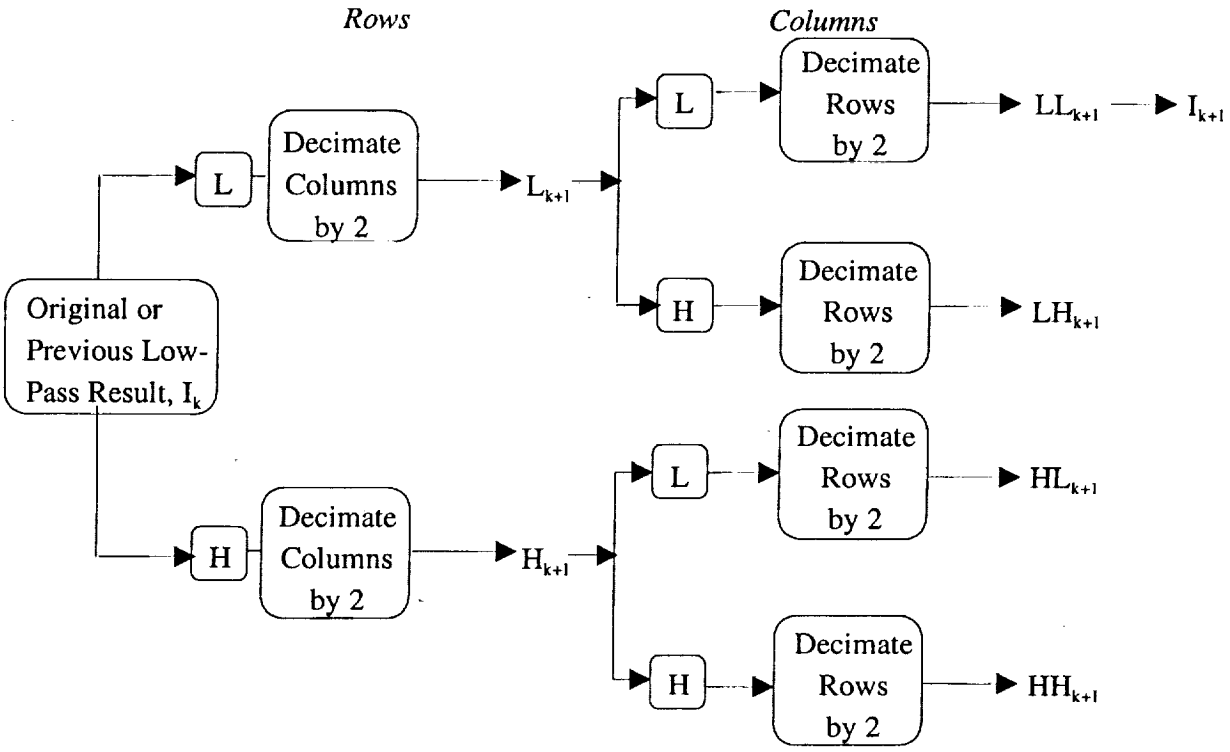

Figure 1
Multi-Resolution Wavelet Decomposition

The input image is first convolved along the rows by the two filters L and H, and the horizontal dimension of these two intermediate results is decimated by 2. Each of the two "column-decimated" images, L_{k+1} and H_{k+1}, is then convolved along the columns by the two filters L and H and decimated along the rows by two. This decomposition results into four images, LL_{k+1}, LH_{k+1}, HL_{k+1} and HH_{k+1}. Each of these images, such as the low/low image, LL_{k+1}, may be taken as the new input to perform the next level of decomposition and so on.

The MRA decomposition algorithm can be described by the following sequence of steps:

(0) Start from the image I_0, level 0 of the multi-resolution sequence (k=0).

(1) High-Pass and low-pass filtering of image rows at level k.

(2) Decimate by 2 the number of columns: results in and L_{k+1} and H_{k+1}.

(3) High-pass and low-pass filtering of image columns at level k.

(4) Decimate by 2 the number of columns: results in LL_{k+1}, LH_{k+1}, HL_{k+1} and HH_{k+1}. The low/low result, LL_{k+1} can be renamed I_{k+1}, since it corresponds to the compression of the original image at level k+1.

(5) Set k to the next level of decomposition, k+1, and continue the iterative process from (1) to (4) until the desired level of decomposition is achieved.

Wavelet reconstruction is obtained by a similar reverse process, which is graphically described in Figure 2, where L* and H* are conjugate filters associated to the previously defined filters, L and H.

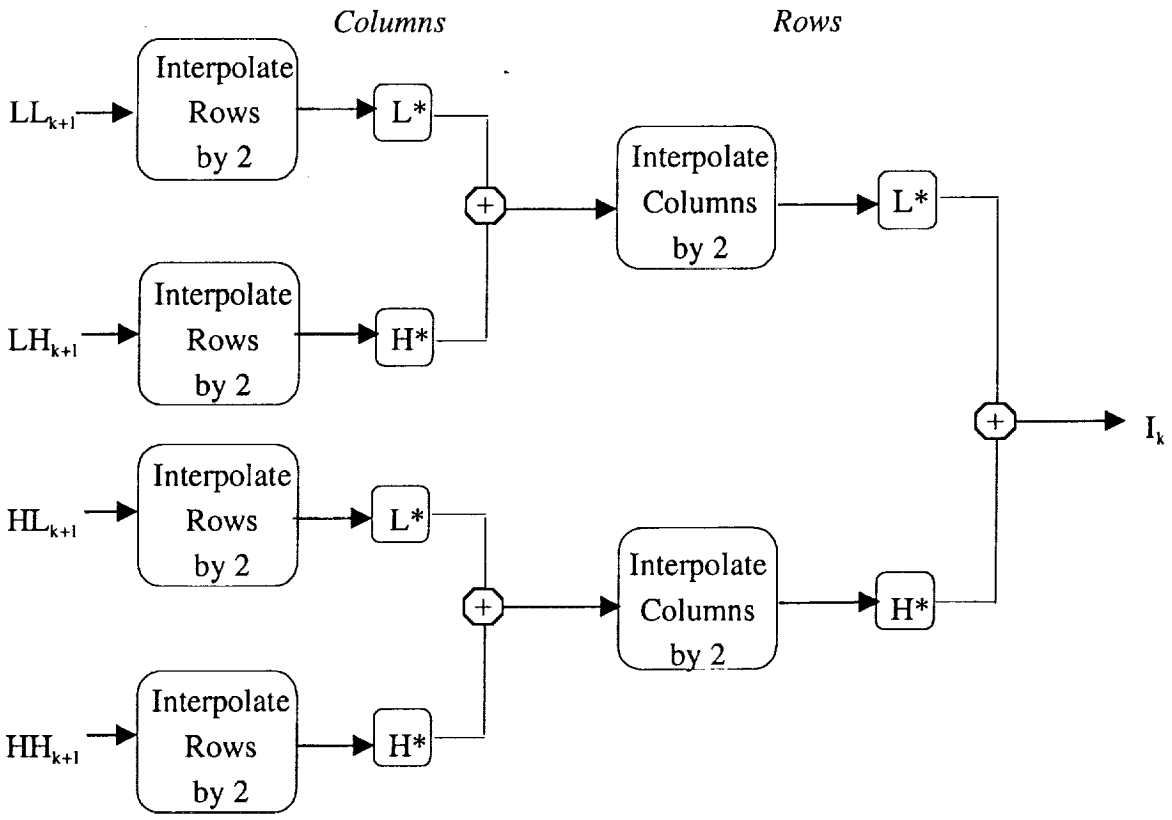

Figure 2
Multi-Resolution Wavelet Reconstruction

3. Overview of the Parallel Systems

Experimental measurements for this work were obtained using the NASA Earth and Space Science (ESS) high-performance computing testbeds. In particular, the NASA HP/Convex SPP-

1000, MasPar MP-2, Hrothgar-Beowulf, and the Jet Propulsions Lab (JPL) Intel Paragon and Cray T3D were used. A brief description of these systems is given below.

3.1 The MasPar

MasPar machines included two families of massively parallel-processor computers, namely the MP-1 and the MP-2. Both systems are essentially similar, except that the second generation (MP-2) uses 32-bit RISC processors instead of the 4-bit processors used in MP-1. The MasPar MP-1 (MP-2) is a fine-grained, massively parallel computer with Single Instruction Multiple Data (SIMD) architecture. The MasPar has up to 16,384 parallel processing elements (PEs) arranged in a 128x128 array, operating under the control of a central array control unit (ACU). The processors are interconnected via the X-net into a 2-D mesh with diagonal and toroidal connections. In addition a multistage interconnection network, called the global router (GR), uses circuit switching for fast point-to-point and permutation transactions between distant processors. A data broadcasting facility is also provided between the ACU and the PEs. Every 4x4 grid of PEs constitutes a cluster which shares a serial connection into the global router. For more information on the MasPar, the reader can consult more specialized MasPar references [12,13].

3.2 The Intel Paragon

The Paragon has a total of sixty-four nodes organized into a 16x4 mesh, of which fifty-four are compute nodes and eight are service nodes. Each node, an Intel GP node, is essentially a separate computer with one compute and one communication i860 processors. Each of the 56 compute nodes has 32 MBytes of memory. The service nodes include: four I/O nodes with 32 MBytes memory and a 4.8 Gbyte RAID each, one HIPPI node with 32 MBytes memory, one User Service node with 32 MBytes memory, and one boot node with 32 MBytes memory and a 4.8 Gbyte RAID. The peak performance (using 56 nodes) is 5.6 GFlops in single precision with an aggregate memory space of 1.8 GBytes and aggregate online disk capacity in excess of 20 GBytes. The programs can be developed in C or FORTRAN which are supported by NX library routines for communication and synchronization purposes.

3.3 The Cray T3D

The Cray T3D is a MIMD system with physically distributed but globally addressed memory. The JPL T3D has a Cray Y-MP as its host system and currently consists of 256 processors each with two MWords (16 MB) of DRAM memory. About 25% of the memory is required by the UNICOS microkernel, therefore, the users can expect to have 12 MB of memory for program and data. Each PE is a 64-bit DEC Alpha microprocessor with a frequency of 150 MHz capable of achieving 150 MFLOPS. The memory interface between the processor and the local memory extends the local

address space to a global address space. The Alpha processor has a direct-mapped data cache organized into 256 lines with 32 bytes per line. Programs can invalidate the local cache as needed to maintain the coherency. Also, remote data entering a processor's local memory can invalidate the corresponding cache line. The system is space-shared through partitions, where the numbers of processors are powers of two. A node consists of two processors sharing a network support logic. All processors are connected by a bi-directional 3-D torus system interconnect network. This topology ensures short connection paths and high bisectional bandwidth. Channels between nodes are two bytes wide and the peak interprocessor communication rate is 300 MB/sec in every direction through the torus. The system software includes FORTRAN (a superset of FORTRAN 77 including many FORTRAN 90 array syntax statements), C, and C++ compilers as well as tools for application performance analysis and parallel code debugging. The PVM is currently supported as are some lower level Cray libraries for passing data and messages among processors.

3. 4 The HP/Convex Exemplar SPP-1000

The HP/Convex SPP-1000 is a distributed-shared-memory multiprocessor. Every eight processors form a hypernode, which is a symmetric multiprocessor. The eight processors of a hypernode are made from four blocks, each with two PA-RISC 7100 processors with 100 MHz clock rate and a 100 MFLOPS peak processing power, 1 MB cache, and 64 MB of RAM. Blocks of a hypernode are interconnected via a 5x5 cross-bar. Hypernodes are, in turn, connected via a scalable coherent

interface (SCI) ring to form a multicomputer. The NASA GSFC SPP-1000 has two hypernodes containing a total of sixteen processors. The Exemplar supports both the virtual memory and the message-passing paradigms. Shared memory is supported via parallelizing compilers that can exploit parallel directives augmented by the user to control the parallel execution. HP/Convex provides compilers for ANSI C, FORTRAN 77, and C++. Message-passing support includes both PVM and MPI.

3.5 The NASA/GSFC Hrothgar Beowulf-Cluster

Beowulf is an architecture for networks of workstations developed at NASA GSFC. The Beowulf philosophy is to use most cost efficient commodity off the shelf (COTS) products for constructing such systems. A Beowulf is basically a pile of PC's interconnected via some LAN technology and running a version of LINUX, a free UNIX, and parallel programming environment such as PVM or MPI. Hrothgar is the specific Beowulf cluster used in this work. The NASA GSFC Hrothgar contains sixteen 100 MHz Pentium processors, each with 16 MB of RAM and 512K cache. The system is interconnected via two fast Eathernet channels, 100 Mbps each. Communication is distributed equally across the channels to provide an aggregate bandwidth of 200 Mbps. LINUX is the underlying operating system and most parallel applications on the system use PVM, although MPI is also supported. See [14,15] for more details on Beowulf clusters.

4. Parallel Implementations

In order to allow accurate measurements of communications, the message-passing programming model was used in all cases, except for the MasPar which used MPL (a data-parallel version of the ANSI C). All message-passing implementations were developed in C and augmented with the appropriate message-passing communication calls. The applications used the "single program, multiple data" (SPMD) programming model. In this model, the same program runs on each node in the application, but each node works on a part of the data. However, because each node is an independent computer, one can also use other programming models. One example is the "manager-worker" model, in which a "manager" program starts up several "worker" programs on other nodes, then gathers and interprets their results.

According to the previous descriptions, the wavelet algorithm can be defined as a combination of successive *filterings* and *decimations*. Our parallel implementation will concentrate on these two operations, focusing on minimizing the communication costs by reducing the number of communication transactions and the distance between the communicating processors.

4.1 The Fine-Grain SIMD Implementations

On the MasPar MP-2, two algorithms were used, referred to as systolic and systolic with dilution; see [9,10] for details. Both of them store the filter in the control unit and broadcast the

filter elements from last to first. After each broadcast, the algorithm requires one multiply and accumulate, followed by shifting the partial result to the left. The algorithm repeats this step for as many times as the size of the filter with partial results being accumulated and built up in a systolic fashion. By the last step, each (logical) processor ends up with one pixel result. The difference between the two algorithms is in the way decimation is handled. In the systolic algorithm, decimation is accomplished using the global router. In the dilution algorithm, the filter is diluted or stretched to be aligned with the relevant pixels, thus avoiding the use of the MasPar global router.

When the image data is larger than the number of the PE's in the machine, a "virtualization" of the PE array has to be defined. Two virtualization methods were considered, "cut and stack" and hierarchical. The hierarchical gave the best results since it improves data locality for the underlying computations [9]. In the "cut and stack" virtualization scheme, the image is cut into squares corresponding to the size of the basic parallel array. For example, if the size of the image is 512x512, we need to stack sixteen layers of image data in the 128x128 parallel array. The hierarchical virtualization divides up the image into sub-images and allocates each sub-image to a different physical processor. The MasPar systolic algorithm was shown to be processor optimal [10].

4.2 The Coarse-Grain MIMD Implementations

Reducing the number of transactions was done by distributing stripes of the image rather than blocks limiting exchange of information to one neighbor instead of two, which would have been needed should image data be distributed by blocks, see figure 3. Secondly, as seen in figure 4, those slices are distributed in a snake-like fashion in order to limit communications to immediate neighbors only. Those communications transactions are needed at the end of each decomposition level in order to build a guard zone around the processor local data from the decomposition results in its neighbors before the next decomposition level starts. Using a striped data decomposition, such zone is only needed for column filtering. In block data decomposition, guard zones need to be established for both the row and column filtering. The depth of the zone is in the order of the filter length. Guard zone data are brought in from the east neighbor for row filtering, and from the south neighbor for the column filter.

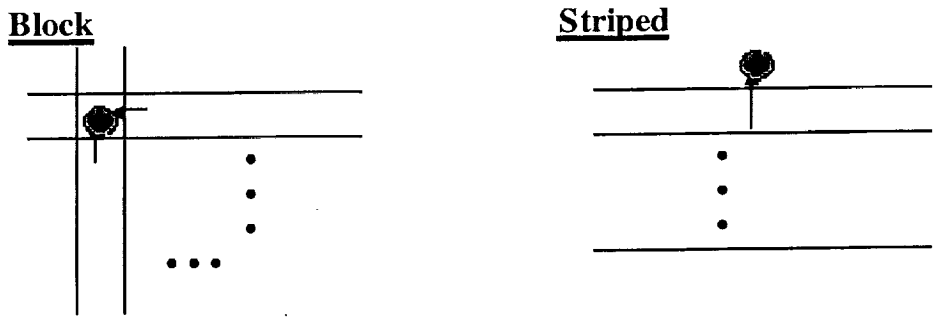

Figure 3
Reducing Communication Transactions Via Striping

The implementations on the Cray T3D, the HP/Convex SPP-1000, and the Hrothgar machines also used a striped approach to minimize the number of communications transaction.

Reducing the number of communications transactions worked better for the used practical filter (guard zone) sizes. This is due to the wormhole routing which amortizes the initial latency cost over larger messages due to its pipelined operation.

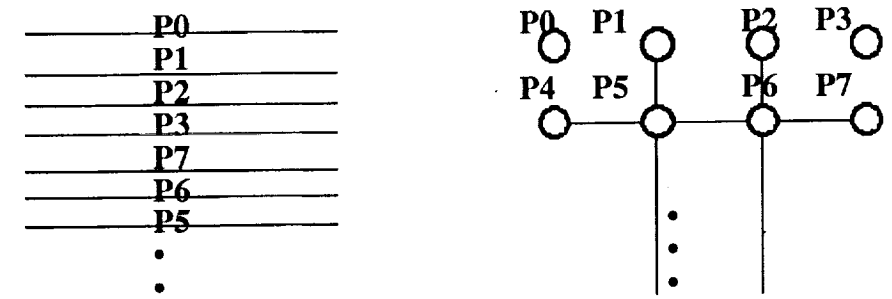

Figure 4

Reducing the Paragon Communications Distances Via a Snake-Like Domain Decomposition

No attempt was made to reduce the communication distances on the T3D. Communication distance is mainly fixed on the SPP-1000 and the Hrothgar due to their cross-bar and the bus architectures, respectively.

5. Experimental Results

Wavelet decomposition of a 512x512 Landsat-Thematic Mapper image of the Pacific Northwest area was used for our experiments (see figure 5). The experimental results for the

wavelet decomposition of this image are given when filters of sizes 8, 4, and 2 are used along with 1, 2, and 4 levels of decompositions, respectively. It should be noted that as the number of decomposition levels increases, more communication is required. Increasing the filter size, however, increases the computational dominance in this problem.

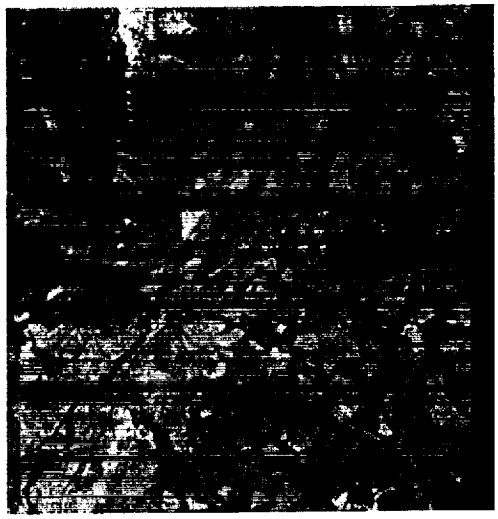

Figure 5

Test Data Included A Landsat Thematic Mapper Image and Different Size Filters

5.1 Intel Paragon Scalability Measurements

The Paragon scaling results are shown in figures 6 and 7. Scalability up to 4 processors was obtained using the straight forward data distribution, where no arrangement was made to limit communication to nearest neighbors. The reason for the number 4 can be seen from figure 4. Beyond 4 processors, processors at the right edge of the network attempt to communicate with those in the leftmost column of the following row. Due to dimension routing, messages in this case travel along the horizontal dimension first before moving along the vertical, which gives

rise to communication conflicts. For the small amount of computations in the wavelet operations, this creates an excessive communications overhead that prevents scalability.

The snake-like data distribution on the other hand does not create these conflicts and limit communication to a distance of one, thus creating the opportunity for relatively better scalability. The Paragon in general, however, shows modest scalability. Communication cost, specially high latency, was observed from the measurements to be the limiting factor still. This can be also noted from figures 5 and 6. With the increase in communications requirements, due to the increase in the levels of decomposition, the speedup curve continues to drop, with the worst case observed at 4 levels.

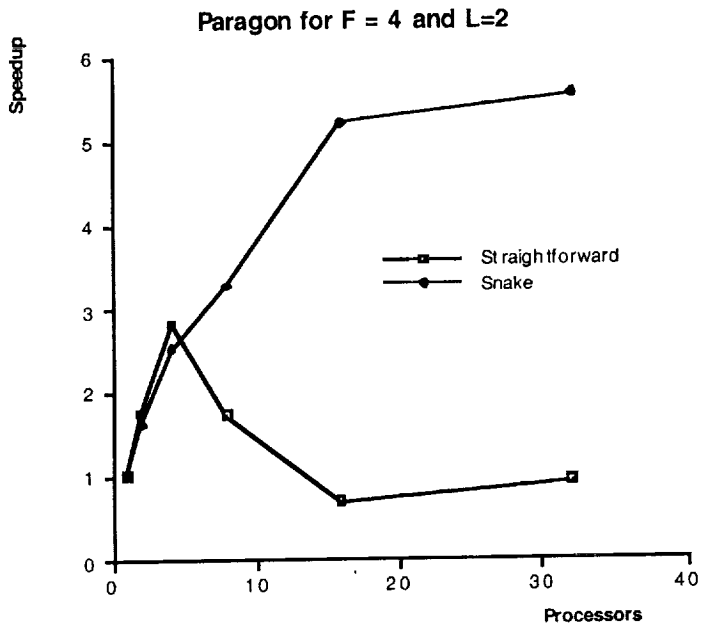

Figure 6

Paragon Performance for Filter Size 4 and 2 levels of Decomposition

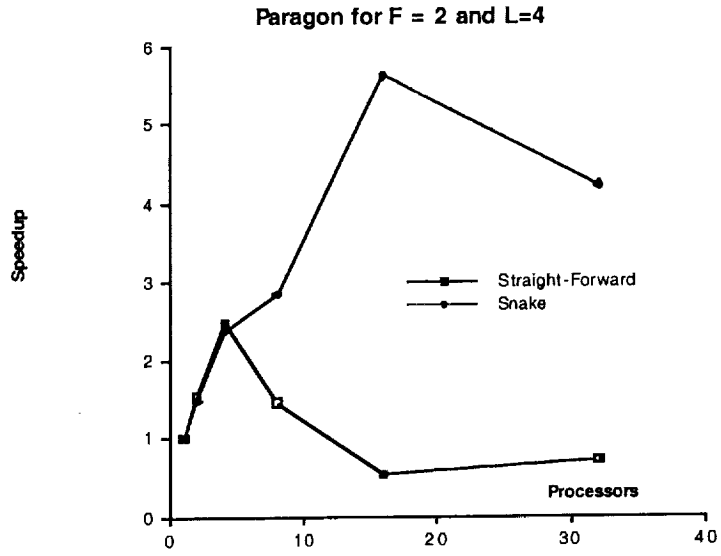

Figure 7
Paragon Performance for Filter Size 2 and 4 levels of Decomposition

5.2 Cray T3D Scalability Measurements

Figures 8 and 9 present the T3D measurements. The Cray, in spite of using the straight-forward data distribution, has shown much better scalability. This has been particularly due to the interconnection network, which is distinguished with its relatively larger degree (degree of 6 in three dimensions) and its very high bandwidth and small latency, when compared to the rest of the used architectures. This is particularly clear from the almost identical scalability results for the Cray in spite of the increase in communications demands.

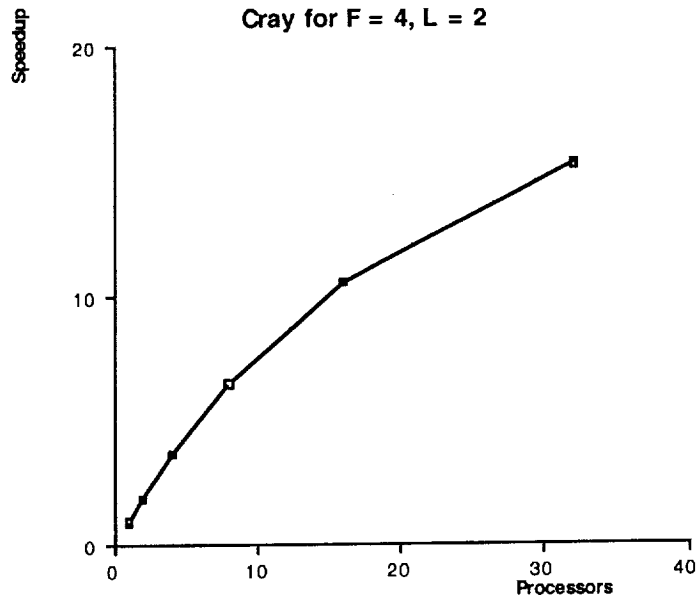

Figure 8

T3D Performance for Filter Size 4 and 2 levels of Decomposition

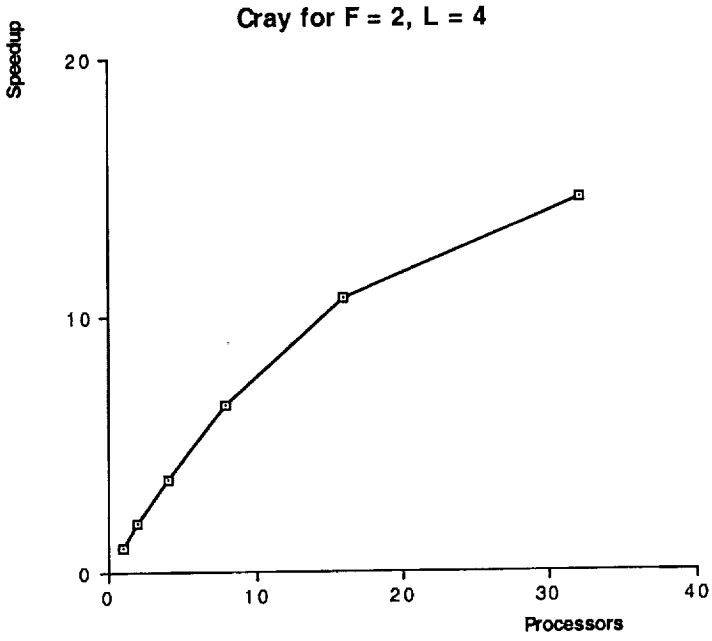

Figure 9

T3D Performance for Filter Size 2 and 4 levels of Decomposition

5.3 HP/Convex SPP-1000 Scalability Measurements

Figure 10 presents the scalability measurements for the HP/Convex SPP-1000 for a variety of filters, and reveals important properties of this architecture, in response to the wavelet image processing workload. In order to put these measurements in perspective, the ideal linear scalability curve (with n processors producing n-fold speed) is plotted on the same axes as the measured cases. The first case, F8/L1, corresponds to a filter of size 8 and one level of decomposition. With one level of decomposition, no communication is necessary, since processors need to exchange data only at the end of one decomposition level in preparation for the next level. With no communication, the speed up curve is expected to be close to the ideal case, but slightly worse, due to parallel overhead other than communication, e.g. redundancy overhead. However, due to the improved caching and infrequent misses, as a result of distributing the image data over multiple processors, a superlinear speed up is observed. Another anomaly is observed for the other two cases. While the scalability is initially close to ideal, and even better than the ideal in the case of F4/2, which requires less communication, the scalability plunges dramatically when the number of processors exceeds eight. In fact, the best performance for these two cases was measured when exactly 8 processors were used. This is due to the fact that for up to 8 processors, the application is distributed among the processors of the same hypernode, and thus is taking advantage of the high communications bandwidth of the 5x5 cross-bar switch. As the number of processors increases beyond eight, additional hypernodes are used

and inter-hypernode communications start to take place over the much slower scalable coherent interface ring.

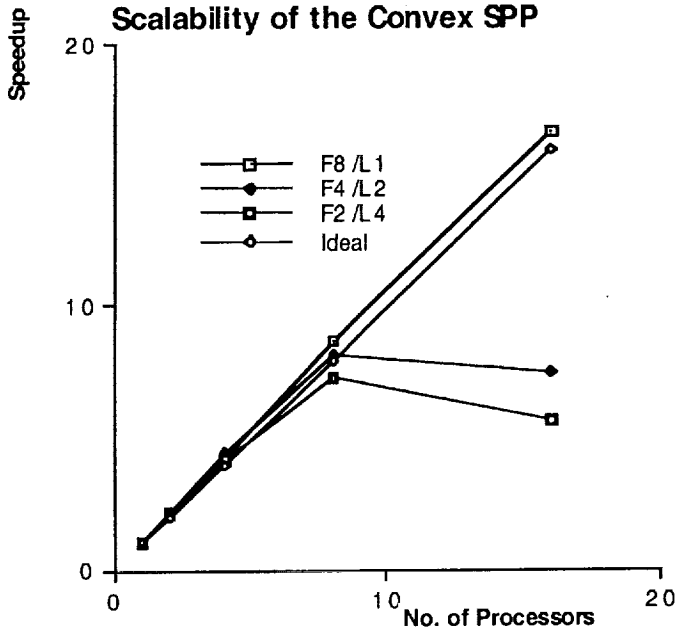

Figure 10
HP/Convex SPP-1000 Performance for Different Filter Sizes and levels of Decomposition

5.4 NASA/GSFC Hrothgar-Beowulf Scalability Measurements

With its 100 MHz Pentium processors and dual 100Mbps Eathernet, Hrothgar seems to have an adequate balance of compute and communication power for the requirements of the wavelet decomposition problem. This is clear from the near linear speedup obtained in figures 11 and 12. An earlier Beowulf generation based on the regular 10 Mbps Eathernet channels and Intel 80486 processors was also used to run the Wavelet decomposition task and have shown very poor

scalability, due to the very low communication bandwidth. Hrothgar represents an improvement, over that earlier system, by 3 folds in processing and 10 folds in communications, which helped making the communication overhead small for wavelet decomposition and, hence, the improved scalability.

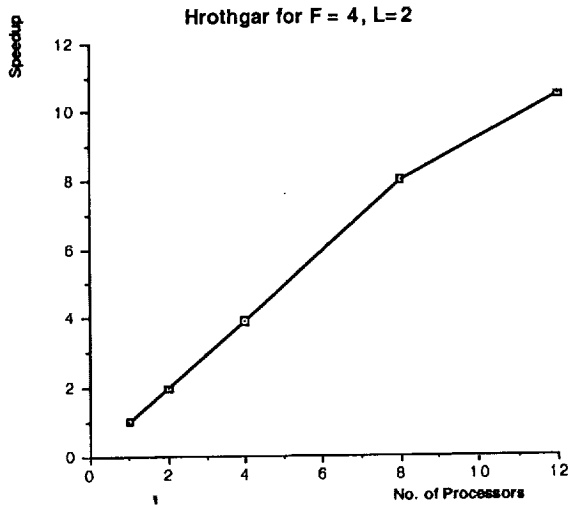

Figure 11

Hrothgar Performance for Filter Size 4 and 2 levels of Decomposition

5.5 Comparative Results

While scalability gives a valuable insight into how balanced and well suited the architecture is for a given application as the number of processors grow, scalability relates the performance of multiple processors of one parallel machine to the performance of one processor from the same machine. Thus, scalability does not report the relative speeds across a number of machines for a given applications. Therefore, the wall clock time to completion for the wavelet

decomposition on the target machines has been measured in order to provide a fair comparative evaluation across the used machines.

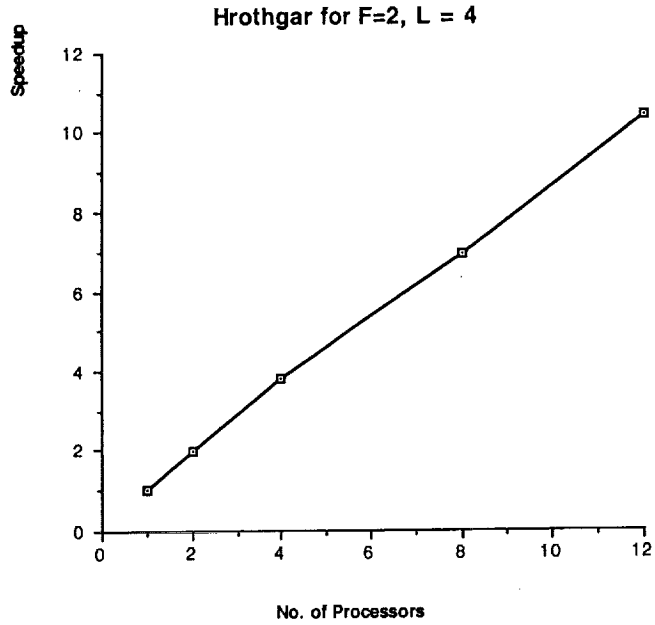

Figure 12
Hrothgar Performance for Filter Size 2 and 4 levels of Decomposition

Table 1 lists these wall clock time measurements in seconds. From the table, it is clear that, for the machine sizes and configurations used, the MasPar is still favorably performing. This is consistent with SIMD machines that have been known to perform well in fine-grain image processing applications. However, the Cray T3D results indicate that MIMD machines that have been only promoted for their general ability can perform well in such fine-grain applications. In fact, for larger image sizes when parallelization overhead is better amortized over more computations, it would be possible for the T3D to do even better. Both the Cray T3D and the MasPar, with the given configuration, are capable of processing 30 images or more per second.

Thus for real-time video, multimedia applications, and scientific and medical applications high-performance computing is quickly asserting its presence. Finally, in spite of its comparatively very low cost, the Hrothgar/Beowulf cluster of PC's has outperformed both the Paragon and the SPP-1000.

Best WCT in seconds for used systems	Filter Size 8 / Levels Decomp. 1	Filter Size 4 / Levels Decomp. 2	Filter Size 2 / Levels Decomp. 4
MasPar (16K)	.0169	.0138	.0123
Cray T3D			
1 processor	.75	.49	.44
16 processors	.05	.03	.0314
Paragon			
1 processor	4.23	3.45	2.78
16 processors	.613	.632	.662
Hrothgar			
1 processor	1.34	1.07	.89
16 processors	.14	.138	.12
CNX SPP			
1 processor	2.28	2.293	2.3
16 processors	.137	.3 (for 8 proc.)	.32 (for 8 proc.)
DEC 5000	5.47	4.54	4.11

Comparative Wavelet Decomposition Performance Measurements
Table 1

6. Conclusion

In this study, we have mapped the multi-resolution wavelet algorithm, developed by Mallat [4], onto several high-performance parallel computers and applied it to remotely sensed data from the NASA Landsat-Thematic Mapper. We have collected an extensive set of performance

measurements for the underlying image processing application over an array of high-performance computers. Both the MasPar and the T3D have provided two orders of magnitude improvement over a workstation, for the specific hardware described here, and can perform wavelet decomposition for video streams in real-time. The Intel Paragon exhibited one order of magnitude improvement and required knowledge about the network operation and special effort to scale beyond four processors. This is greatly attributed to the relatively low communication bandwidth and latency when compared with the processing power. The HP/Convex SPP-1000 could not scale for the used data sizes beyond 8 processors due to the excessive overhead associated with communicating over the scalable coherent interface ring. When no communications was required, the large cache on the SPP-1000 has resulted in a superlinear speedup. The performance of the Cray T3D almost did not change when the communication requirements were increased, exhibiting good scalability. Surprisingly, the Hrothgar/Beowulf network of PC's has compared favorably in timing with the SPP-1000 and the Intel Paragon. Such Beowulf architecture clearly compares favorably with all used massively parallel architectures on performance/cost basis. As image sizes become large, MIMD machines are expected to do at least as good as SIMD in such traditionally fine-grain applications as image processing. This is due to the expected amortization of parallel overhead when the problem size increases, and which would lead to operating these MIMD structures at a much higher efficiency than observed with 512x512 images. Both types of architectures, however, have demonstrated

their ability to meet the requirements posed by real-time video and NASA remote sensed scientific databases.

References

[1] C.K. Chui, *An Introduction to Wavelets, Wavelet Analysis and its Applications*, Volume 1, Academic Press, 1992.

[2] J.P. Djamdji, A. Bijaoui and R. Maniere, "Geometrical Registration of Images: The Multiresolution Approach," *Journal of Photogrammetry and Remote Sensing*, Vol. 59, No.5, May 1993, 645-653.

[3] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1992.

[4] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, " *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, July 1989.

[5] G. Strang, "Wavelets and Dilation Equations: A Brief Introduction," *SIAM Review,* Vol. 31, No. 4, pp. 614-627, December 1989.

[6] J. LeMoigne, W.J. Campbell, and R.F. Cromp, "An Automated Parallel Image Registration Technique Based on the Correlation of Wavelet Features," *IEEE Transactions on Geoscience and Remote Sensing,* in press, 2001.

[7] M.A.Cody, "The Fast Wavelet Transform," *Dr.Dobb's Journal,* April 1992.

[8] *Numerical Recipes,* Chapter 13.10, "Wavelet Transforms," pp 591-606, 1992.

[9] A.K. Chan, C. Chui, J. Le Moigne, H.J. Lee, J.C. Liu, and T.A. El-Ghazawi, "The Performance Impact of Data Placement for Wavelet Decomposition of Two-Dimensional Image Data on SIMD Machines," *Frontiers'95, Fifth Symposium on the Frontiers of Massively Parallel Computation*, McLean, VA, Feb. 6-9, 1995, pp. 246-251.

[10] T. El-Ghazawi and J. Le Moigne, "Multi-Resolution Wavelet Decomposition on the MasPar Massively Parallel System," *International Journal of Computers and Their Applications*, September 1994.

[11] T. El-Ghazawi and J. Le Moigne, "Wavelet Decomposition on High-Performance Computing Systems," *Proceedings of the 25th International Conference on Parallel Processing (ICPP'96)*, Bloomingdale, IL, CRC Press, August 1996.

[12] *MasPar Technical Summary.* MasPar Corporation, November 1992.

[13] J. Nickolls, The Design of the MasPar MP-1: A Cost Efficient Massively Parallel Computer, *Proceedings of the IEEE Compcon*, Feb. 1990.

[14] T. Sterling, D. Becker, D. Savarese, J.E. Dorband, U.A. Ranawake, and C.V. Packer, "Beowulf: A Parallel Workstation for Scientific Computation," *1995 International Conference on Parallel Processing*, Aug. 1995.

[15] T. Sterling, P. Merkey, and D. Savarese, "Improving Application Performance on the HP/Convex Exemplar," *IEEEC*, December 1996.

[Fan89] Z. Fang, X. Li, and L. M. Ni, On the Communication Complexity of Generalized 2-D Convolution on Processor Arrays, *IEEETC*, February 1989.

[Fan86] Z. Fang, X. Li, and L. M. Ni, Parallel Algorithms for 2-D convolution, *Proceedings of ICPP'86.*

[Pfief95] W. Pfiefer, S. Hotovy, N. Nystrom, D. Rudy, T. Sterling, and M. Straka, "JNNIE: The Joint NSF-NASA Initiative on Evaluation," *SDSC Tech. Report GA-A22123*, San Diego, July 1995.

[Lee87] S.-Y. Lee and J. K. Aggarwal, Parallel 2-D Convolution on a Mesh Connected Processor Array, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* July 1987.

[Lee94] H.J. Lee, J.C. Liu, A.K. Chan, and C.K. Chui, Parallel Implementation of Wavelet Decomposition/Reconstruction Algorithms, *Proceedings SPIE Wavelets'94,* Orlando, April 5-8, 1994.

[Lu93] J. Lu, Parallelizing Mallat Algorithm for 2-D Wavelet Transforms, *Information Processing Letters*, 45, 255-259, 1993.

[Mar89] M. Maresca and H. Li, Morphological Operations on Mesh Connected Architectures: A Generalized Convolution Algorithm, *Proc. of the 1986 IEEE CS Conf. on Computer Vision and Pattern Recognition,* pp 299-304.

[Sto83] Q. F. Stout, Mesh-Connected Computers with Broadcasting, *IEEETC*, September 1983.