Performance Evaluation of Multidimensional Transpose Parallel FFT Algorithm on PC Cluster

Humayun Baig Meerja and Sirajuddin Shaik
Department of Information and Computer Science
King Fahad University of Petroleum and Minerals
KFUPM#506, Dhahran 31261, Saudi Arabia
E-Mail: {humayun, siraj}@ccse.kfupm.edu.sa

Abstract

In this paper, we present the performance evaluation of multidimensional transpose parallel Fast Fourier Transform (FFT) algorithm on pc cluster, which is basic component in applications such as the pseudospectral methods. This algorithm is implemented and analyzed on 8-processor cluster by taking 3D FFT as an example. Communication is based upon the standard portable Message Passing Interface (MPI). We have shown that processor scaling for execution time at fixed problem size can be obtained provided that transpose algorithm is optimized for simultaneous block communication. We have also shown that for the large problem sizes speed up and efficiency is good.

Keywords: FFT, Pc cluster, MPI, speedup, efficiency, communication time

1. Introduction

A large variety of FFT algorithms are available to scientists that use the conventional serial or vector computers. However for parallel machines, the choice of FFT algorithms is very limited. In applications such as the pseudospectral methods for solving partial differential equations (PDE's), a number of multidimensional FFT's are computed per time step. The other scientific and technical applications in which FFT used are Time Series and Wave Analysis, solving Linear Partial Differential Equations, Convolution, Digital Signal Processing and Image Filtering etc. The speed of the FFT computation is therefore very critical to any large application using the pseudospectral method. Since such very large computations are feasible mostly on only parallel machines using MPI [1] , there is a need for fast multidimensional FFT algorithms for parallel machines.

The approach to computing multidimensional FFT's on parallel machines is currently under debate. There are two methods that are possible [2][3]. One of the approaches is the

``Transpose Method". In this method, planes between nodes divide data. For example, in the three-dimensional transform, each node has a number of planes on which it computes two-dimensional FFT's. Next, a distributed transpose rearranges the data in such a way that the FFT along the third dimension can be computed locally. The parallel aspect of this approach is limited to the distributed transpose, which is equivalent to a standard exchange problem [3]. Here, each node sends data to and receives data from all other nodes during distributed transpose. This method is fairly easy and has been implemented for a number of applications.

The organization of material in this paper is as follows: Section 2 briefly describes and discusses transpose parallel FFT algorithm for 3-dimention FFT, overview of the used cluster architecture and the methodology used during the experiment. Section 3 analyzes the performance of the transpose algorithm. Section 4 ends with conclusion.

2.0 Methodology:

2.1 The generalized Transpose Algorithm

In the two-dimensional transpose algorithm, the input size n is arranged in an $\sqrt{n}$ x $\sqrt{n}$ two-dimensional array that is portioned along one dimension on p processes.

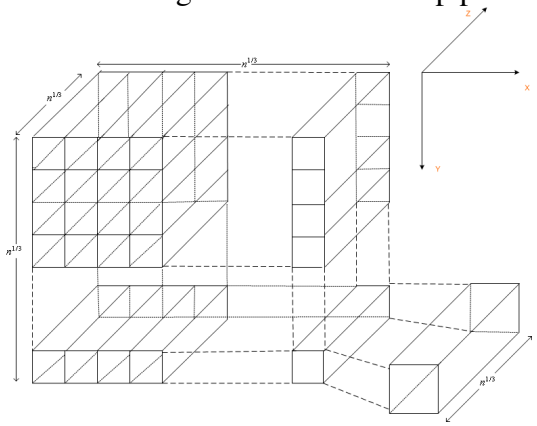

Figure 1: Data distribution in the three-dimension transpose algorithm for n-point FFT on p processes

As the extension of this scheme, the n data points to be arranged in a $n^{1/3} \times n^{1/3} \times n^{1/3}$ three-dimension array mapped onto a logical $\sqrt{p} \times \sqrt{p}$ two-dimension mesh of processes.figure1 illustrate this mapping. From the figure 1 each process has ($n^{1/3}$ / $\sqrt{p}$) x ($n^{1/3}$ / $\sqrt{p}$) x $n^{1/3}$ =n/p elements of data .In general, the three-dimensional transpose algorithm works in five phases:

1. In the first phase, $n^{1/3}$ –point are computed on all the rows along the z-axis.
2. In the second phase, all the $n^{1/3}$ cross-sections of size $n^{1/3x}$ $n^{1/3}$ along the y-z plane are transposed. (See Figure 2 (a)).

Top

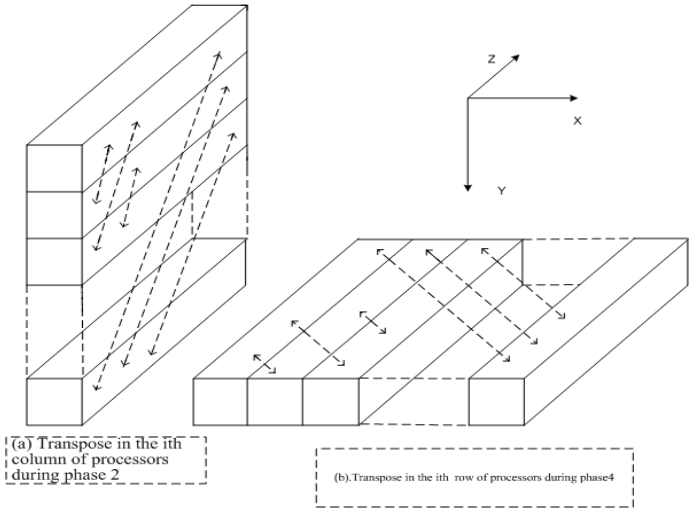

Figure 2: Three Dimensional transpose Algorithm

3. In the third phase, $n^{1/3}$ –point FFT's are computed on all the rows of the modifies array along the z-axis.
4. In the fourth phase, $n^{1/3}$x $n^{1/3}$ cross-sections along the x-z plane is transposed (See Figure 2 (b)).
5. In the fifth and final phase, $n^{1/3}$-point FFT's of all the rows along the z-axis are computed again.

2.2 Environment Setup

The performance analysis was implemented and ran on a cluster of PCs with the hardware and software configuration shown in the Table 1.

Hardware	8 PIII nodes
Processor Speed	1.0 GHz
Memory	2.0 GB / node
Interconnection	100 Mb/s Fast Ethernet
Operating System	Linux 2.4.9-31
Parallel Environment	MPICH
Compiler	gcc 2.96

Table 1: PC cluster configuration

2.3 Hypothesis:

During this experiment, we assume all our input sequences to be real numbers with no complex numbers and all operations are in real numbers. We also assume the complex root of unity (*W*) equals to one all the way during the experiment.

Top

3.0 Evaluation and testing

3.1 Communication and Computation time

The performance of 3D FFT is under optimal conditions are compared for varying problem sizes. This is shown in the Figure 3 Figure 4 Figure 5 respectively. Test cases are shown for three problem sizes N=64,128,256 .the processors are varied from 1 to 8.A fully switched network with optimal message passing is employed.

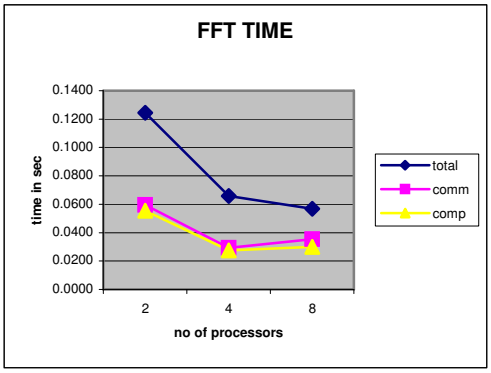

Figure 3: FFT times (total computation, communication) for different processors for the size 64.

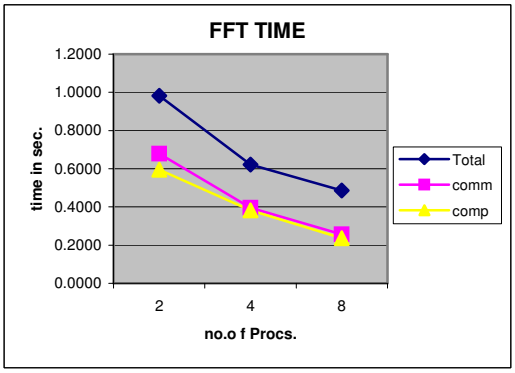

Figure 4: FFT times (total computation, communication) for different processors for the size 128.

Top

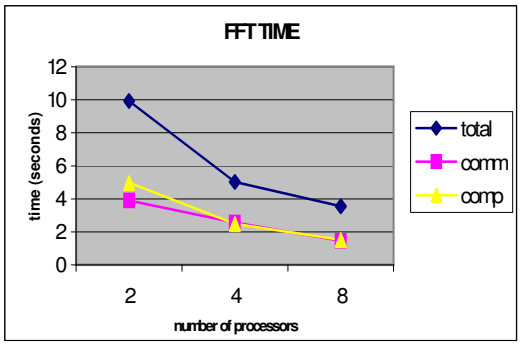

Figure 5: FFT times (total computation, communication) for different processors for the size 256.

The total time shown in Figure 3 Figure 4 Figure 5 is broken into computation and communication times. As the number of processors increases total time taken by the 3d FFT is decreases gradually. The communication and computation times are also almost decreases as the number of processors increases and some times communication and computation times are same .so the scaling with processors works as expected especially at the large number of processors.

3.2Efficiency

The efficiency is the ratio of total time on one processor to P times the P processor total time [4]. The efficiency of the 3D FFT for different no of Processors and for different problem sizes is shown in the Figure 6.

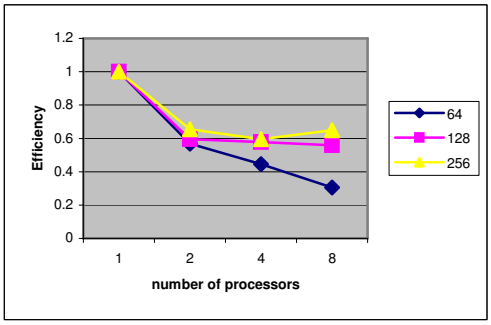

Figure 6:3D FFT efficiency as a function the number of processors for sizes 64,128, 256.

It is apparent from the Figure 6 that efficiency is better for larger problem sizes.

3.3 Speedup

Speedup is ratio of computational time on single processor to the computational time on n processors. The projection speedup on different number of processors for 3D FFT for problem sizes 64,128,256 is shown in Figure 7. From the figure it is clear that as the number of processor increases the speedup gradually increases for larger problem sizes.

Top

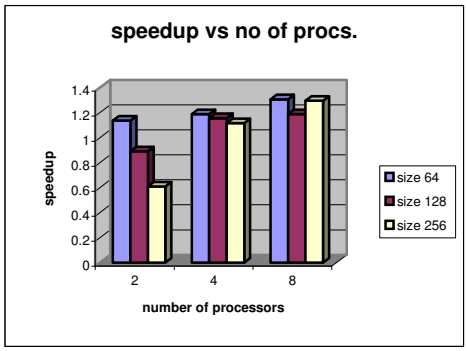

Figure 7:3D FFT speedup projections on problem sizes 64,128,256.

4.0 Conclusion

We have discussed the transpose algorithm for 3D FFT using MPI on distributed memory multiprocessor computer. Using this algorithm, communication is arranged for non-overlapping pair-wise communication between the processors, due to which we achieved high efficiency and speedup (lower computation time) for multi- dimensional FFT's.

5.0 References

[1] William Gropp, Ewing Lusk, and Anthony Skjellum (1994) Standard portable Message Passing Interface (MPI).
[2] M. Quinn. Parallel Computing: theory and practice. (1993) McGraw Hill Higher Education.
[3] V. Kumar, A. Grama, A. Gupta, and G. Karypis.(1994) Introduction to Parallel Computing, design and analysis of algorithms. Addison Wesley Higher Education.
[4] P.Dmitruk, L.P.Wang, H.Matthaeus, R.Zhang, D.Seckel, Scalable Parallel FFT for spectral simulations on a Beowulf cluster.

Acknowledgment

We would like to thank King Fahad University of Petroleum and Minerals for its support and resources to carry out this work.

Top