# Parallel Matrix Transposition Using Stream Programming Paradigm

Saeid Yousefpour [1], Azad Shojaei [2*], Rahim Rashidi [3]

[1] Department of Computer,Boukan Branch, Islamic Azad University, Boukan, Iran
[2] Department of Computer, Saghez Branch, Islamic Azad University, Saghez, Iran
[3]Department of Computer, Boukan Branch,Islamic Azad University, Boukan, Iran
*Azad.Shojaei@gmail.com

**Abstract:** Parallel computing has become the dominant paradigm in computer science, mainly in the form of multi core processors. One software approach for parallel programming, is writing programs in streaming model. A stream program is type of computer program such that the input data is stream of data. In this paper we have used stream programming model, and developed the parallel version of matrix transposition algorithm. Our source codes are simple java code and used JStream java library. Matrix transposition is a fundamental matrix operation of linear algebra and arises in many scientific and engineering applications. For example, matrix transposition is one of the major tasks in image and signal processing and matrix decompositions. Also matrix transposition is a permutation frequently performed in various techniques involving systems of liner equations. Partial differential equations are typically solved using the Alternating Direction Implicit (ADI) method by transposing the data between the solution phases in different directions. Another example in which data transposition may be advantageous is solving Poisson's problem using the Fourier Analysis Cyclic Reduction (FACR) method. Also our code shows that writing parallel programs in streaming model would be very simple.

[Saeid Yousefpour, Azad Shojaei, Rahim rashidi. **Parallel Matrix Transposition Using Stream Programming Paradigm.** *J Am Sci* 2012;8(7):471-475]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 73

**Key words**: Data Stream; Stream Programming; Parallel Computation; Parallel Matrix Transposition; Multi Core

## 1. Introduction

Matrix transposition is a fundamental matrix operation of linear algebra [1][2] and arises in many scientific and engineering applications. For example, matrix transposition is one of the major tasks in image and signal processing and matrix decompositions. Also matrix transposition is a permutation frequently performed in various techniques involving systems of liner equations. Partial differential equations are typically solved using the Alternating Direction Implicit (ADI) method by transposing the data between the solution phases in different directions [3]. Another example in which data transposition may be advantageous is solving Poisson's problem using the Fourier Analysis Cyclic Reduction (FACR) method [4].

On a uniprocessor, an algorithm involving a transposed matrix may not actually require the matrix data to be transposed in physical memory. Instead, it may be accessed simply by exchanging the row and column indices. However in a distributed memory multiprocessor environment, we cannot simply interchange the global row and column indices. Instead, the data must be physically moved from one processor to another. On the other hand, the problem of writing efficient code for parallel computers or compiling for such architectures is notoriously difficult, such that there emerged the so-called "Parallel software crisis". Modern microprocessor technology has even decided to solve part of this problem in hardware by analyzing data and control dependencies at run time and dispatching instructions to pipeline and parallel functional units on-line, although this automatic reordering at runtime is only applicable within the limited scope of a small window containing a few subsequent instructions in the code [5].

But one other software approach, is writing programs is streaming model, which called stream processing. The stream processing paradigm simplifies parallel software by restricting the parallel computation that can be performed. Given a set of data (a stream) a series of operations (computational units) are applied to each element in the stream. Speech encoding, image processing and signal processing programs are some examples of stream programs [6] [7]. Due to the nature of stream programs, they can be easily mapped to distributed or multi core architecture [8]. The structure of each program forms a graph, the nodes are computational units and the edges are data paths. This is called stream graph.

Conceptually a stream computation is a data transfer in the program. The units that data is transferred between them are called filters. A filter is a computational unit such that in each execution time it reads one or more data from input channel and after processing sends it to the output channel. Filters are independent from each other and they contain their own code there is no global variable or any reference to the other filters. In a stream program connection between filters, makes a graph that output of some filters is connected to the input of other filters [6].

The structure of stream graph is generally constant during the execution of program. That is a certain set of computational units are repeatedly applied in a regular, predictable order to produce an output stream that is given function of the input stream [5]. There are many compilers and tools for developing stream programs. Some examples are Cg [9] [10], StreamC [11] [12], Brook [13], StreamIt [14] [15] [16] [17] [18] and JStream Java library [19].

In this paper we have used JStream library to develop parallel version of matrix transposition algorithm. More details on JStream can be found in [19]. The rest of this paper is organized as follow: section 2 is related works on parallel matrix transposition problem. Section 3 is a review of JStream java library. Section 4 explains our work and matrix transposition algorithm in stream programming model and section 5 is conclusion.

**2.Related works**

Transposition of a matrix is a redistribution of its elements. Many researchers have considered the problem for different architectures. Eklundh considered the problem of directly accessing rows or columns of a matrix when its size is larger than the available high-speed storage. The data should be stored on an external storage device, allowing direct access. The performance of the algorithm depends on the size of the main storage, which at least should hold 2n+1 point. In that case the matrix has to be read in and out n times [21].

O'Leary, implemented an algorithm for transposing an N*N matrix on a one-dimensional systolic array. This architecture uses $n^2$ switching processors and $n^2$ bit buffers. Arrays are also given to take a matrix in by rows and put it out by diagonals and vice versa [22]. Azari, Bojanczyk, and Lee [23] developed an algorithm for transposing an M * N matrix on an N * N mesh-connected array processor. Johnsson and Ho presented an algorithm for a Boolean n-cube or hypercube. Their algorithms make use of the processing elements (PE's) in parallel [24]. Tsay, Ding and Wang present an algorithm on mesh that supports wormhole switching [4]. Choi, Dangarra and Walker presented algorithm based on the block scattered decomposition. The communication schemes of the algorithm are determined by the greatest common divisor (GCD) of the number of rows and columns of the processor template [25].

**3.A review on JStream Library**

The JStream library has base classes for filters and communication patterns. The library supports three communication patterns: Pipeline, Split-Join, Feedback loop. Pipeline is the simplest form of communication models.

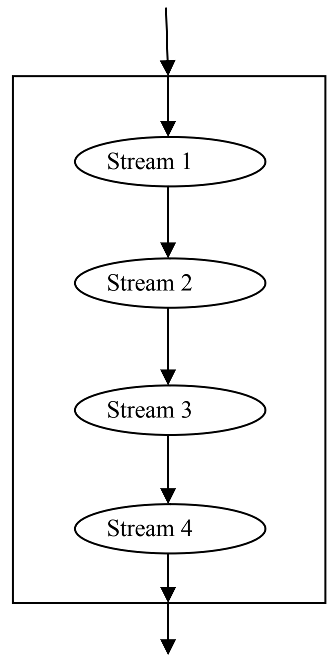


**Figure1.**Pipeline model of filters communication[19]

A pipeline has a number of child streams. In this structure of filters, the output of first stream is connected to the input of second stream and the output of second stream is connected to the input of third stream and so on [19]. Figure 1 shows a pipeline structure.A feedback loop structure has a body stream. The output of body stream is sent to a splitter. One branch of splitter leaves the loop and another branch is returned back to the body through a joiner. The joiner joins the input channel of feedback loop with loop channel. Data type of input channel and output channel and loop branch should be the same [19][20]. Figure 2 shows a feedback loop structure.

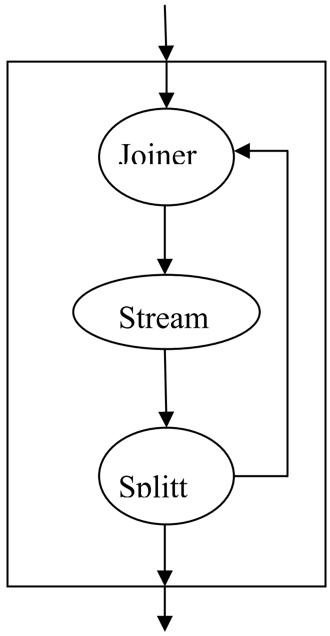


**Figure 2**. Feedback loop model of filters communication[19]

A split-join structure enables parallel processing. Each split-join stream has a number of child streams that we can define them. In the split-join every input data is distributed to every child and after process the results is gathered and sent to the output channel [20]. Figure 3 shows a split-join structure.

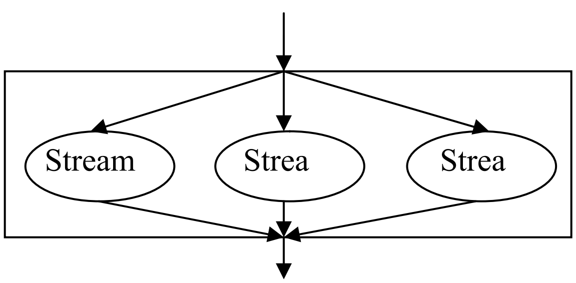


**Figure 3.** Split Join model of filters communication[19]

In a split-join incoming data passes through a splitter, is distributed to the child streams for processing and then is fed through a joiner to be recombined into a single output stream. The split-join has two splitter, duplicate and round robin splitter. Duplicate splitters take each incoming item and push the same item to each of the child stream, duplicating data. Round robin splitters take each item and send it to exactly one of the child streams, in order.

Splitroundrobin() causes one item to be sent to each output, in order; splitrounribin(2) causes two items to be sent to the first stream, two to the second and so on. Round robin joiners are identical to round robin splitters, except that they read from the input streams in specified pattern and write data to the output stream. As mentioned above, authors in [19] explains these classes and patterns in more details.

**4. Matrix transposition with stream programming model**

In order to transpose a matrix we have to use a split join structure. We need a filter that passes input data to the output (without any computation). We called this filter *passing filter*. If the matrix is n*m, then there should be m passing filter as split join child streams. There for the number of split join children must be equal with the number of matrix columns**.**

The type of splitter and joiner is round robin splitter and round robin joiner. The splitter distributes each data item in rows to each child. The round robin joiner collects all data of each child. Figure 4 shows the method.

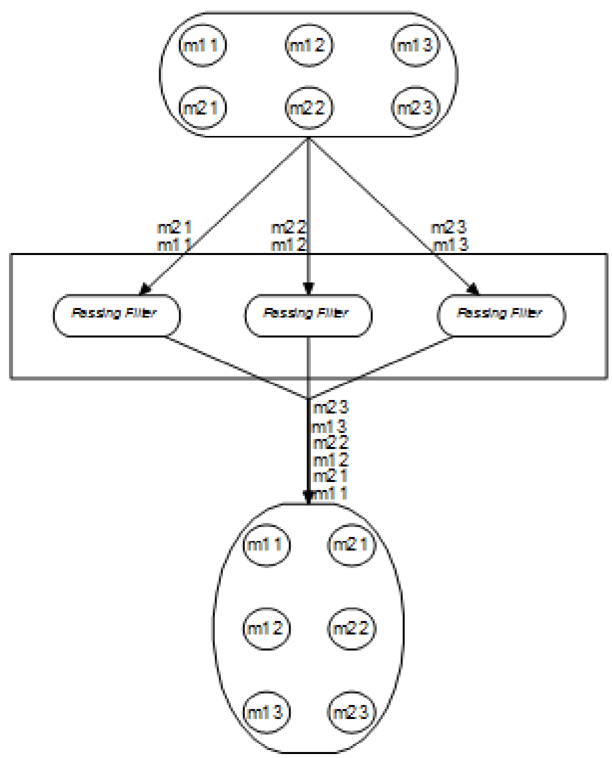


**Figure 4** - Matrix transpisition method

Figure 5 shows the stream graph for matrix transposition. Matrix Reader is a filter that reads matrix elements and Matrix Printer is a simple filter that prints the transposed matrix.

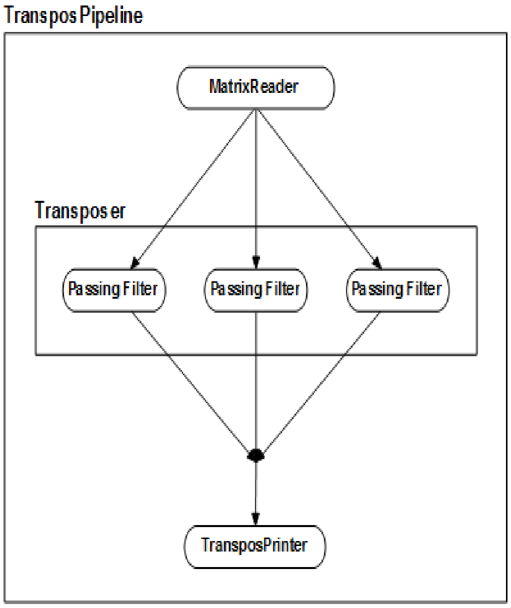


**Figure 5** . Stream graph for matrix tranposition

The code (1) shows the passing filter.

```
import JStream.*;
public class PassingFilter extends
Filter<Integer,Integer>
{
        public PassingFilter()
        {
                Pop=2;
        }
        public static PassingFilter
PassingFilterConstruct()
        {
                PassingFilter Obj=new
PassingFilter();
                return Obj;
        }
        public void run()
        {
                Push(Pop());
                Push(Pop());
        }
}
```

*Code 1 – Passing Filter*

Code (2) shows split joins JStream Code.

```
import JStream.*;
public class Transposer extends
SplitJoin<Integer,Integer>
{
        public Transposer()
        {
                SplitRoundRobin();
                Add(PassingFilter.
PassingFilterConstruct());
                Add(PassingFilter.
PassingFilterConstruct());
                Add(PassingFilter.
PassingFilterConstruct());
                JoinRoundRobin(2);
        }
        public static Transposer TransConstruct()
        {
                Transposer Obj=new Transposer();
                return Obj;
        }

}
```

*Code 2 – Split join Construction for matrix transposition*

## 5. Conclusion

Parallel computing is a form of computation in which many calculations are carried simultaneously. It is now well established that parallel computing is moving into the mainstream with a rapid increase in the adoption of multi core processors. Unlike previous generations of mainstream hardware evolution, this shift will have a major impact on existing and future software.

In this paper we have developed parallel matrix transposition algorithm with stream programming paradigm. Stream programs can be easily mapped to distribute or multi core architecture. In our algorithm, we construct a simple passing filter that passes data items without any computations and one split join structure. If our matrix is n*m matrix then we have to have m passing filter as child of split join. The splitter and joiner type is round robin. Also because of using java library, there is no need to learn other parallel language or syntax.

**Corresponding Author:**
Azad Shojaei
Department of computer, saghez Branch,
Islamic Azad University, Saghez, Iran.
E-mail: Azad.Shojaei@gmail.com

## References

[1] Golub, G.H and Van Loan, C.V, 1989. Matrix Computations. The Johns Hopkins University Press, Baltimore, MD. Second Edition.

[2] Strang, G,1988. Linear Algebra and Its Applications. Harcourt Brace Jovanovich, Inc., San Diego, CA, Third Edition.

[3] Johnsson, S. L and HO, C.T, 1990. Optimizing tridiagonal solvers for alternating direction methods on Boolean cube multiprocessors, SIAM Journal on Scientific and Statistical Computing, Vol. 11, pp. 563-592

[4] Tsay, J.J., Ding K. S and Wang W.T, 2003. Optimal Algorithm for Matrix Transposing on Wormhole-Switched Meshes, Journal of Information Science and Engineering, Vol. 19, pp. 167-177.

[5] Christoph W. Kessler., Parallelism and Compilers, 2000. MSc Theses, Trier University.

[6] William Thies, 2009. Language and compile support for stream programs, PhD theses, MIT.

[7] Elliot, L and Waingold, SIFt, 2000. A Compiler for Streaming Applications, MSc Theses, MIT.

[8] Jasper, H. S., Jean Privat., Rechid Guerraoui and Jan Vitek, 2007. StreamFlex: High-Throughput Stream Programming in java, Proceedings of the 22nd annual ACM SIGPLAN conference on Object-Oriented programming systems and applications, OOPSLA07, Montreal, Canada.

[9] William, R., Mark, R., Steven Glanvile., Kurt Akeley., Mark, J and Kilgard, Cg, 2003. A system for Programming Graphics Hardware in C-Like Language. Proceedings of the International Conference on Computer Graphics and Interactive Techniques, San Diego, California.

[10] Randima Fernando and Mark J. Kilgard, 2003. The CG tutorial: the definitive guide to programmable real-time. Addison Wesley, Boston, USA.

[11] Abhishek Das, William J. Dally and Peter Mattson, 2006. Compiling for stream processing, proceedings of the l5 international conference on parallel architectures and compilation techniques, Seattle, Washington, USA.

[12] Xuejun Yang, Ying Zhang, Jingling Xue, Ian Rogers, Gen Li, Guibin Wang, Exploiting loop-dependent stream reuse for stream processors, proceedings of the 17th international conference on parallel architectures and compilation techniques, New York, USA, 2008.

[13] Shin-Wei Lio, Zhaohui Du, Ganasha Wu and Guei-Yuan Lueh, 2006. Data and computation transformation for brook streaming applications on multi processors, proceedings of the international symposium on code generation and optimization.

[14] Streamlt: a language for streaming application, online resource: http: csail.mit.edu.

[15] Michel Karczmarek, William Thies., Saman Amarasinghe,2003. Phased scheduling of Stream programs, proceedings of the 2003 ACM SIGPLAN conference on language. Compiler and tool for embedded systems, San Diego, California, USA.

[16] Michel I. Gordon., William Thies, Saman Amarasinghe, 2006. Exploiting coarse-grained task, data, and pipeline h parallelism in stream programs, proceedings of the l2 international conference on architectural support for programming languages and operating systems, San Jose, California, USA.

[17] Andrew A. Lamb., William Thies and Saman Amarasinghe,2003. Linear analysis and optimization of stream programs, San Diego, California, USA.

[18] Streamlt language specification, online resource; http: csail.mit.edu.

[19] Yousefpour, S and Mahjur, Ali, 2010. Object Oriented Stream Programming with Java, proceedings of the 2010 international conference on intelligent network and computing (ICINC 2010), kuala Lumpur, Malaysia.

[20] Rashidi, R and Yousefpour, S, 2011. Parallel Bubble Sort Using Stream Programming Paradigm, The 5th International on Application of Information and Communication Technologies, Baku, Azarbayjan.

[21] Eklundh, J. O, 1972. A Fast Computer Method for Matrix Transposition. IEEE Transactions on Computers, Vol. 21, pp. 801-803.

[22] O'Leary, D. P. 1987. Systolic Arrays for Matrix Transpose and Other Reordering, IEEE Transactions on Computers, Vol. 36, pp. 117-122.

[23] Azari, N. G., Bojanczyk, A. W., and Lee, S. Y, 1988. Synchronous and Asynchronous Algorithms for Matrix Transposition on MCAP. In SPIE Vol. 975, Advanced Algorithms and Architecture for Signal Processing III, pp. 277-288.

[24] Johnsson, S. L., and Ho, C. T, 1988. Algorithms for Matrix Transposition on Boolean n-Cube Configured Ensemble Architecture. SIAM J. Matrix Anal. Vol. 9, pp. 419-454.

[25] Choi, J., Dongarra J. J and Walker, D. W,1993. Parallel Matrix Transpose Algorithms on Distributed Memory Concurrent Computers, Rice University.


6/6/2012