University of Witwatersrand

ELEN4020

Data Intensive Computing In Data Science

Laboratory Exercise 2

Authors:
Kishan Narotam - 717 931
Jesal Chana - 603 177

Authors:
Syed Hussain - 600 524
Yusuf Ally - 604 973

9th March 2018

I. Problem Description

A square matrix, defined as $A[N_0][N_1]$, must be created with various dimensions, where $N_0 = N_1 = 128, 1024, 8192$. The matrix could be an integer which is 4 bytes or a short floating point which too is 4 bytes. The elements of the matrix must be generated using the formula presented in equation 1:

$$A\langle i,j\rangle = i * N_j + j \tag{1}$$

The transpose of these matrices must be computed without wasting memory by creating a copy of the original matrix, but rather modify the original matrix to become the transposed matrix. Since the size of the matrix is so large, PThread and OpenMP programming must be performed on the matrix. The number of threads that must be used are $4, 8, 16, 128$ for each value of N_0. The time taken to perform the transposition of each matrix must be recorded.

II. Background

A. PThread

A thread is a set of instructions that are independent and scheduled to run by the operating system. A POSIX Thread (PThread) however is defined as a standardized programming interface for UNIX systems, i.e. an execution model independent of any programming language. It could also be defined as a set of *C* language programming types, functions, procedures and constants that are implemented with the library header file, `pthread.h` [1].

PThreads are also regarded as low-level application programming interface (API) when managing threads. It allows the user to have as refined control as possible over the threads as well as multiple exclusion objects [2]. Since the Pthreads are such low level, it is fairly limited to the language used.

B. OpenMP

Open Multi-Processing (OpenMP) is an API that is used explicitly to direct multi-threading and shared memory parallelism. It is comprised of three primary API components:

- Compiler Directives
- Runtime Library Routines
- Environment Variables[3].

OpenMP is a much higher level API, and portable allowing the user to utilize this API in different languages such as C, C++ and Fortran. It is more easily scaled than PThreads as it has the ability to divide the work across multiple threads with ease [2], [3].

For this type of problem, the program that utilizes OpenMP should reveal faster times than the program that utilizes PThreads.

C. Amdahl's Law

Amdahl's Law is a key concept that is used to understand in parallel computing. It is used to predict the maximum speedup in latency for a program processing using multiple processors [4].

In parallel computing, the law states that if P is the program that can be made parallel, i.e. proportion of execution time, and given that $1 - P$ is the remaining proportion, i.e. the proportion that is still serial, the maximum speedup in latency using N number of processors can be defined as equation 2.

$$S_{latency} = \frac{1}{(1-P) + \frac{P}{N}} \tag{2}$$

The speedup is limited by the total time needed for the serial proportion of the program. For example is a program need 10 hours using a single processor, and 9 hours can be parallelized, and 1 cannot, the maximum speedup is limited to $10\times$ using equation 2 [4].

III. Function Description

IV. Pseudo Code

V. Final Code and Output

VI. Different Methods

A. Splitting the Matrix

Since parallel computing can be used, another method that can be used to transpose a matrix would be to split the matrix into smaller matrices and giving each thread a smaller matrix to transpose. Figure 1 gives an example an 8×8 matrix that is split into four smaller matrices and 4 threads would be applied to this. Each thread handles a smaller 4×4 matrix and processes the matrix and calculates the transpose.

Although an efficient method, this would be poor on the memory. The reason for this is that an index would have to be created in order to track which elements have already been transposed. This is a trade off of this method, where efficiency is improved however memory is wasted.

$$\left[\begin{array}{cccc|cccc}
a[0][0] & a[0][1] & a[0][2] & a[0][3] & a[0][4] & a[0][5] & a[0][6] & a[0][7] \\
a[1][0] & a[1][1] & a[1][2] & a[1][3] & a[1][4] & a[1][5] & a[1][6] & a[1][7] \\
a[2][0] & a[2][1] & a[2][2] & a[2][3] & a[2][4] & a[2][5] & a[2][6] & a[2][7] \\
a[3][0] & a[3][1] & a[3][2] & a[3][3] & a[3][4] & a[3][5] & a[3][6] & a[3][7] \\
\hline
a[4][0] & a[4][1] & a[4][2] & a[4][3] & a[4][4] & a[4][5] & a[4][6] & a[4][7] \\
a[5][0] & a[5][1] & a[5][2] & a[5][3] & a[5][4] & a[5][5] & a[5][6] & a[5][7] \\
a[6][0] & a[6][1] & a[6][2] & a[6][3] & a[6][4] & a[6][5] & a[6][6] & a[6][7] \\
a[7][0] & a[7][1] & a[7][2] & a[7][3] & a[7][4] & a[7][5] & a[7][6] & a[7][7]
\end{array}\right]$$

Fig. 1. An 8×8 matrix that is split into smaller 4×4 matrices

VII. Conclusion

The conclusion goes here.

References

[1] Barney, B; Livermore, L; *POSIX Threads Programming*; https://computing.llnl.gov/tutorials/pthreads/; Last Accessed: 01/03/2018

[2] Ball, M; *c - Pthreads vs. OpenMP - Stack Overflow*; https://stackoverflow.com/questions/3949901/pthreads-vs-openmp; Last Accessed: 01/03/2018

[3] Barney, B; Livermore, L; *OpenMP*; https://computing.llnl.gov/tutorials/openMP/; Last Accessed: 01/03/2018

[4] Unknown; *What is Amdahl's Law? - Definition from Techopedia*; https://www.techopedia.com/definition/17035/amdahls-law; Last Accessed: 02/03/2018