Evaluation of computational efficiency by the method of parallelization in a multi-core CPU’s in a large data array

F A Alisherov¹, S Q Iskandarov²,⁴, S I Khamraeva³ and B B Nurmetova³

¹Founder of Algorismus Ltd, Amir Timur, 24a, Tashkent, Uzbekistan  
²Phd researcher, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Amir Timur, 108, 100200 Tashkent, Uzbekistan  
³Urgench branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, 110, al-Khwarizmi str., 220100, Urgench, Uzbekistan  
⁴E-mail: sanjariskandarov@gmail.com

Abstract. Nowadays, with the rapid growth of data, there is a growing demand for large-scale data processing. This data processing, led to the development of computers. These include the development of computer processors, cores. This article explores the effectiveness of Microsoft framework-based parallel extension for multi-core processors for fast Fourier Transform (FFT) in fast multi-faceted computing paradigms. We have a very large number of threads as a multi-thread paradigm, and the cost of switching between threads is minimal. The global nature of computational and transactional equilibrium and built-in communication models in the parallel FFT algorithm makes it an ideal candidate for multidimensional platforms.

1. Introduction

In the development of a software tool, in the process of developing the object architecture of the program, the information models of the objects that the user wants to work directly are selected and their software specification is completed, as well as their user interface is defined. Such object classes or individual tactical objects form architectural network systems (subsystems). This network provides interoperability between systems. When using active objects in the approach to the object, the main broad class of architectures is a set of programs that run in parallel, with the same abstract objects performing the function of programs. The client-server architecture is a good example of such a class architecture. In such a system, one of the active objects, called a "server", provides certain software services at the request of other active objects, called a "client" [1]. These types of services, in turn, form the following groups that allow virtualization of data classifications between networks.

Parallel computing systems - perform calculations on a computer or software, using multiple nodes. If a computer processes large amounts of data only through hardware components, the computer relies only on physical cores, and if the software focuses on increasing the efficiency of processing large amounts of data, the computer relies on logical cores.

Execution of IA-32 (the first set of x86 to support 32-bit computers) uses the execution resources of processors, an average of 35%, and 65% of the execution resources of processors are empty, which is inefficient from the capabilities of the processor indicates use. The operation of the processor should be carried out in such a way that it is necessary to make maximum use of its capabilities in each clock cycle. This is exactly the idea that Hyper-Threading technology implements by using the free resources of processors to perform parallel tasks.
It is the integration of two or more cores in a single microprocessor to perform the task more efficiently in parallel. Such a single-core multi-core configuration provides the highest rate of inter-core wear in multi-processor systems than in external buses, switches, and so on. A multi-core architecture consists of a complete functional set of two or more resources to increase the performance of processors [2].

2. Literature review

Scientists Okonta O E at el who are currently conducting research in parallel to perform large-scale data array processing and increase the efficiency of large-scale data flow processing. Indicators have shown an increase in performance by up to 30% [3, 4].

MPI (Message Passing Interface is a standardized and messaging standard developed by a team of academic and industry researchers to work on different parallel computing architectures) and Open MP (Open MPI - the best available MPI libraries) in solving various problems in M K Buza's research. Set of experience, technology and resources for) studied the problem of the effectiveness of the use of technology. Although these technologies work with different types of memory organization (distribution and sharing), the possibility of sharing MPI + open MP technologies to develop parallel applications was explored, its effectiveness was evaluated, and the use of technologies together (hybrid) was more effective [5].

Zaynidinov X.N., Xamdamov U.R. In their research, they developed parallel algorithms based on data streaming models using parallel programming libraries, such as Open MP and TBB (Threading Building Block). Haar and Dobeshi have studied the use of discrete wavelet conversion algorithms to speed up signal processing. It has been proven that the performance of the algorithm also depends on the L2 cache size in multi-core processors, because the currents generated in the processor cores share this memory [6].

Musaev M M, Berdanov U. In order to increase the efficiency of parallel computing in the research of A, it is necessary to analyze computational schemes, to divide them into small tasks, to determine the order of their small data exchange tasks and share them with each processor core. and at the same time in multi-core processors it is explained that the division into very large parts does not allow to load the processors equally and to achieve the minimum computing time [7].

3. Algorithm to use parallel loops and increase efficiency

The Fast Fourier transform (FFT) algorithms compute the Discrete Fourier Transform (DFT) while reducing a complexity of N2 to NLog2 (N). Consider a complex array of N values; the raw DFT is given by the following formula:

\[ X(f) = \sum_{k=0}^{N-1} x_k e^{-2\pi i kf/N} = \sum_{k=0}^{N-1} x_k W_N^k \quad (1) \]

\[ W_N = e^{-2\pi i/N} \quad (2) \]

The FFT algorithms refactor this formula, so they compute the same values as the DFT except for possible rounding errors. They can be used independently or combined providing several trade-offs in terms of computational complexity, memory requirement and parallelism [11].
Based on the above, the issues of increasing the efficiency of machine calculations for any computer are relevant. The use of parallel loops, driven by threading technology, can be developed based on the following algorithm.

Fast Fourier Transform (FFT) is one of the most important signal processing and data analysis algorithms.

Parallel Extension include main three components:

1. Task Parallel Library (TPL), we can express the parallelism in the existing sequential code, means we can express the code as the Parallel task, will be run concurrently on all the available processors [8].

2. PLINQ is a query execution engine that accepts any LINQ-to-Objects or LINQ-to-XML query and automatically utilizes multiple processors or cores for execution when they are available. The change in programming model is tiny, meaning you do not need to be a concurrency guru to use it [9].

3. Coordination Data Structures-A set of structures used to synchronize and coordinate the execution of parallel tasks [10].

The article provides an analysis of the results of the program developed based on the Parallel. For() method, which is mainly one of the materials of TPL. Parallel. For Method (System.Threading.Tasks), when the thread pool's heuristics is unable to determine the right number of threads to use and could end up injecting too many threads. For example, in long- The Max Degree of Parallelism property

---

**Figure 1.** A combined hybrid block diagram of fast binary-inversion and FFT algorithms
affects the number of concurrent operations run by Parallel method calls that are passed this Parallel Options instance. To use Parallel.For(), the computational process is required to consist of unrelated parts. If the result of one part of the calculations is related to another part of the calculations, it is recommended that such calculation processes be calculated on a distributed basis or on a conveyor basis.

4. Discuss and result
Let's look at this using the following example to calculate the number of steps involved in multiplying matrices and the time it takes to multiply them. Here we can evaluate the possibilities of Parallel for based on threading. Using a parallel for loop looks like this: To evaluate the calculations, let us look at the multiplication of matrices. The following two computers were used to evaluate linear and parallel computing, the parameters of which are given in table 1.

| Parameters        | Computer N1                                      | Computer N1                                      |
|-------------------|--------------------------------------------------|--------------------------------------------------|
| **Model**         | HP Laptop 17-by0xxx                               | HP Z6 G4 Workstation                             |
| **OS**            | Windows 10 pro 64-bit                             | Windows 10 Pro 64-bit (10.0, Build 18363)        |
| **Processor**     | Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (8 CPUs) | Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz (24 CPUs), ~2.1GHz |
| **RAM**           | 8072MB RAM                                       | 131072MB RAM                                     |

The results of the calculation of large data streams in computers in the processing of large data arrays and the time for the computational process in each machine sample are shown in table 2. It is also possible to estimate the level of computer resource utilization in the calculations.

| Number of operations | Computer N1 Serial calculation(ms) | Computer N1 Parallel calculation(ms) | Computer N2 Serial calculation(ms) | Computer N2 Parallel calculation(ms) |
|----------------------|------------------------------------|-------------------------------------|-----------------------------------|-------------------------------------|
| 356400               | 2                                  | 54                                  | 2                                 | 8                                   |
| 5,970,000            | 30                                 | 45                                  | 24                                | 10                                  |
| 47,920,000           | 219                                | 226                                 | 189                               | 32                                  |
| 83,895,000           | 432                                | 401                                 | 332                               | 55                                  |
| 239,760,000          | 1170                               | 528                                 | 1039                              | 124                                 |
| 419,650,000          | 2412                               | 808                                 | 1996                              | 211                                 |
| 2,638,680,000        | 16890                              | 7776                                | 13722                             | 1454                                |
| 5,038,320,000        | 33216                              | 16906                               | 24754                             | 2731                                |
| 17,995,500,000       | 169747                             | 61653                               | 119889                            | 11377                               |
Figure 2. A comparative diagram of time in linear and parallel computing

5. Conclusion
It can be seen from the graphs given that while the use of threading technology in small calculations is considered somewhat inefficient when the capacity of the computer is high, if the number of calculations increases, this figure will change to the efficient side. The study found that parallelism in multi-core processors increased efficiency by 2.98 times for the 1st computer and by 10.5 times for the 2nd computer. It is also possible to increase the efficiency based on the computational volume when performing calculations by creating a virtual machine for arbitrary complexes of the machine.

References
[1] Reindeers J 2007 Understanding task and data parallelism. (Topic:Tech & Work)
[2] Michael H 2017 Concurrency Basics. (University of Maryland: Department of Computer Science)
[3] Okonta O E, Ajani D, Owolabi A A, Imiere E E, Uzomah L 2015 Performance Evaluation of Hyper threading Technology Architecture Using Microsoft Operating System Platform. (West African Journal of Industrial & Academic Research) 15(1)
[4] Saini S, Mehrotra P, Taylor K, Aftosmis M, and Biswas R Performance 2011 Analysis of CFD Application Cart3D Using MPInside and Performance Monitor Unit Data on Nehalem and Westmere Based Supercomputers 13th IEEE Intl. Conf. on High Performance Computing and Communications
[5] Buza M K 2015 Analysis of the effectiveness of parallel technologies. ISSN 1561 – 5359 Piece of Intellect 1(2)
[6] Zainidinov H N, Khamdamov U R 2019 Algorithms for parallel wavelet transform of signals on multi-core processors. TUIT Bulletin Tashkent University of Information Technologies named after Muhammad Al-Khwarizmi
[7] Musaev M M Berdanov U A 2016 “The Technology of Parallel Processing on Multicore Processors” (International Journal of Signal Processing Systems), 4(3) pp 252–257
[8] Dot Net Tutorials official web page https://dotnettutorials.net/lesson/task-parallel-library-overview/
[9] Microsofts official web page for extantion https://docs.microsoft.com/en-us/archive/msdn-magazine/2007/oktober/parallel-linq-running-queries-on-multi-core-processors.
[10] Popov D V, Kulikov A N 2012 Parallel computing using MS NETFREAMWORK 4.0 parallel extension for example FFT. Multi-core processors, parallel programming, FPGA, signal processing system pp 7–14
[11] Julien H at el 2015 Implementation of a Fast Fourier Transform Algorithm onto a Manycore
Processor. Conference on Design and Architectures for Signal and Image Processing (DASIP)