MANAGEMENT REVIEW

Parallel Image Processing Based on Superscalar Pipeline

Biswa Ranjan Acharya

Apex Institute of Technology & Management
Joholo, Pahala, Bhubaneswar

Abstract

This paper presents a parallel image processing model based on pipeline, concerning the current efficiency of remote sensing image processing, which cannot meet the need of processing increased number of remote sensing data. In the multi-core CPU era, the pipeline technology can process the complex and time-consuming steps of the remote sensing image processing in superscalar parallel. This model can improve the processing efficiency significantly. The principles of the pipeline and the parallel computing are illustrated in detail in this paper. The obtained experimental results show that the parallel processing model based on pipeline is efficient in the processing of remote sensing image with large amount of data.

Keywords

Remote sensing image processing; Multi-core processors, pipeline computing, parallel computing

Introduction

In recent years, remote sensing and earth observation technology make new progress continuously. We receive a large number of new geospatial data from many different sensors every day. If we can't process these data on time, they will lose their value. With the development of the remote sensing, the remote sensing technology is gradually applied to more and more industries, such as cartography, precision agriculture, environmental monitoring, military reconnaissance and so on [1]. As to the need of the national economic development, a stable operating system will be built, the self-sufficiency rate of spatial data will be improved and a industry chain of spatial information will be formed to 2020 [2].For these reasons, fast data processing algorithms and techniques that can speed up data processing become more and more important. Currently, there are already many ways to speed up the processing of remote sensing image. But these methods are most based on excellent hardware foundation. The existing parallel computing is also many computer clusters approach. Although this method is achieved satisfactory results, it is too expensive. The problems is that there are rarely methods to speed up the remote sensing image processing only based on the existing hardware and sophisticated algorithms. In the computer information age, multi-core processor which can perform more tasks in a period of time by its unique technological advantages is the development trend of future processor. However, it is a disappointment that we can't achieve desired results in enhancing data processing speed if we only use multi-core processors without efficient parallel algorithms. Pipeline is a commonly used parallel processing techniques which can fully use the advantages of multi-core processors. We can use pipeline combined with the existing algorithms to develop parallel programs for remote sensing image processing. These programs can improve the efficiency of data processing, and it will be a good way for processing the vast amount of remote sensing data. This article will explore the implementation of using the pipeline technology to achieve parallel processing of remote sensing image. The lee filtering on a remote image will be taken as an example in the experimental part to verify the treatment effect.

Ii. Parallel Processing Model Based On Pipeline

A. Parallel Computing

Parallel computing, simply, is the calculation made on parallel computers, it often said that high-performance computing, because parallel technology is essential to any high-performance computing and supercomputing invariably [6]. Parallel computing can be divided into time-parallel and space-parallel. The former typical representative is pipeline technology, however, the later do simultaneous computing with multiple processors [7]. This article focuses on the pipeline model on multi-core processors for the remote sensing image processing. Multi-core processor integrates two or more complete computing cores, each of which is essentially a relatively simple microprocessor. Multiple computing cores can execute instructions in parallel so that can achieve a chip thread-level parallelism, and it can also perform more specific tasks within the time to achieve task-level parallelism [3]. This will increase processor's computing power obviously. In a single computer, if you want to do parallel computing you have to use the parallel program which can use the resources of the multi-core processor to improve efficiency. With the current development of parallel computing technology and the increasing popularity of parallel programming, in the foreseeable future, a large number of application development programmers will be asked to write parallel programs. This trend also raises new demands for the study of parallel programming [4].

B. Pipeline Processing Model

The pipeline is a vivid description of the instruction execution in the CPU, and it is also an implementation mechanism which can make the processor overlap to execute the instructions [9]. The main job of CPU is to take instructions, interpret and execute instructions. Instruction pipeline has been widely used in the computer.

Suppose the system cost the same time to take and execute one instruction and the duration is marked as m t, then under normal circumstances, a complete implementation cycle time will be two m t. If the pipeline is used here, the taking of the instruction and the executing of the instruction can be implemented at the same time, so the total execution time will remain as m t, the efficiency has be doubled . In order to achieve greater efficiency in the implementation, the pipeline can be divided into more stages:
instruction fetch (IF), instruction decoding (ID), executing (EX), memory (MEM), write back (W), five stages in total. Suppose the time spent in each stage is equal.

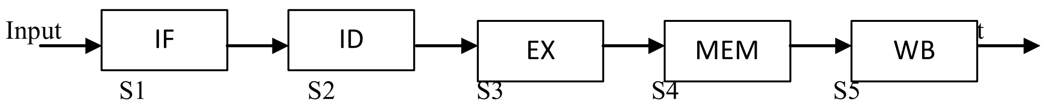

Figure 2. Instruction pipeline

Assume that the costing time of each sub-process is m t, the space-time diagram of instructions flowing through the pipeline can be showed as follows:

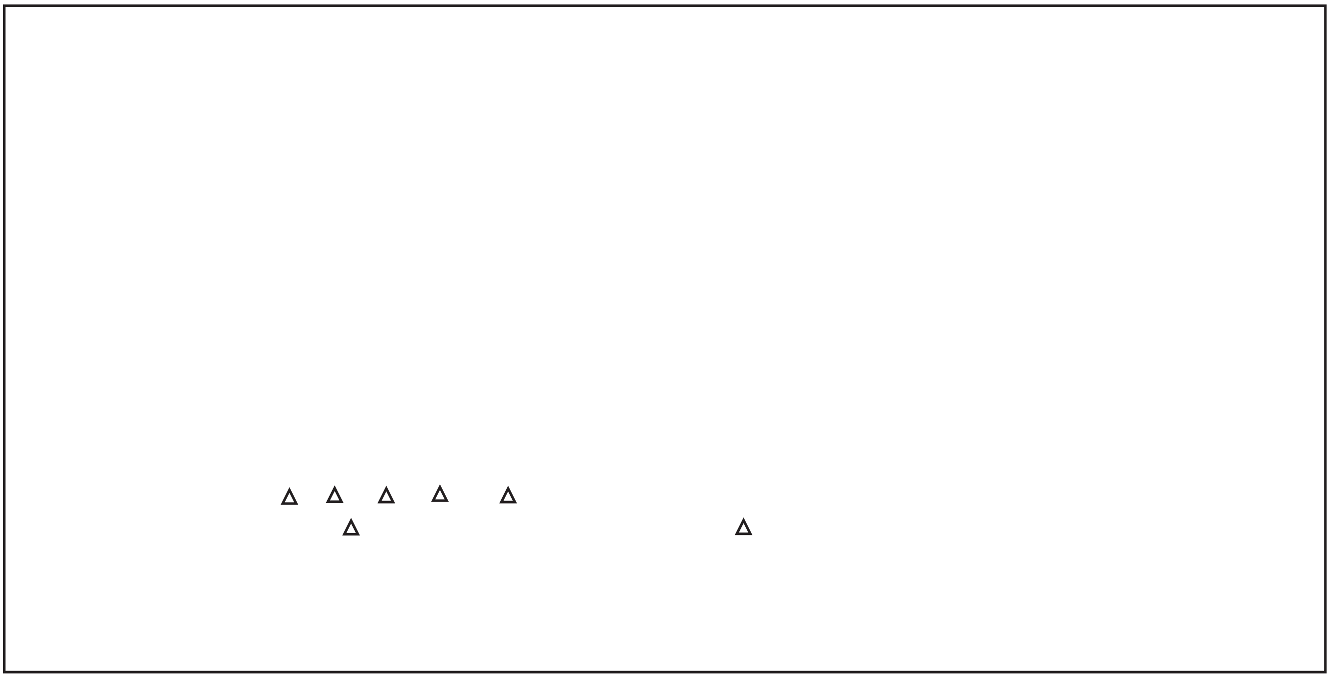

Figure 3. Space-time diagram

Through using the pipelining, multiple instructions can be executed in a quasi-parallel way that is the program overlap to operate the instructions. This is why the pipeline technology can improve the efficiency of instruction execution. Form the programmer's point of view, there are two advantages of using software pipelining. The one is that pipeline provides a way to solve complex problems in program, the other is that software pipelining can re-use of different procedures [5]. The software pipelining can also improve the performance of programs and the CPU utilization [10]. So, if a large amount of remote sensing image data is divided into a certain size block of data suit for the computer memory, the pipeline technology can be used to improve the efficiency of processing.

III. Remote Image Processed in Pipeline

Currently, most remote sensing image processing algorithm is applied to serial procedural. Even though the existing processing models can satisfy some of the data processing requirements, it is difficult to improve the processing speed further. What's more, the progress of the study on the more efficient algorithms can't meet the need of processing the massive data. For the common large grid image, it can be cut into small blocks based on certain rules. Take a 1000 * 1000 pixel image for example, if we take the upper left corner as the origin and cut a length of 100 pixels along the horizontal and vertical, then one hundred 100*100 pixel blocks we will get. These blocks can be processed in parallel by pipeline. So, a pipeline with three stages can be created to process the image. One stage is used for inputting blocks, one is used for processing and the other one for writing the processed blocks back into files on the hard disk. As the disk has only one head of limitations, the reading and writing operations of the image blocks can only be implemented in serial. However, parallel processing mechanism can be used in the processing stage to improve efficiency, because this stage is just needed to do some specified processing to the blocks flowing in and there are no relations among the different blocks. Assuming that the processing time of each stage is equal, the parallel processing model based on pipeline is shown in Fig.4.

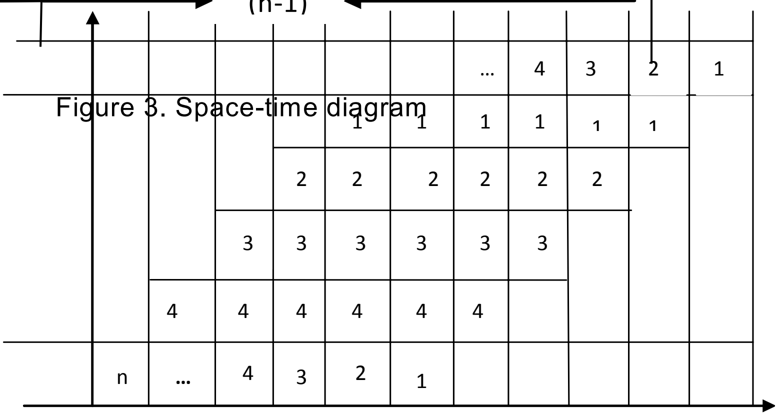

Figure 4. Parallel process on pipeline

The pipeline performance can be measured by the formula:

$$S = T1/Tp \quad (1)$$

The T1 is the time spent in serial procedure of a remote image processing algorithm, and Tp is the time spent in parallel procedure of the same algorithm. There are three main factors affecting the performance of the software pipeline based on certain hardware [9]. The first one is parallelism. Clearly, in a serial stage, each token must be processed serially in order. In a parallel stage, multiple tokens can be processed in parallel by the stage. If the parallelism was too high, there might be a problem where the unordered stage in the middle keeps gaining tokens because the output stage cannot keep up. This situation typically leads to undesirable resource consumption by the middle stage. If the parallelism was too low, the system can't give full play to the advantages of parallel. Both of these cases will affect the performance of the pipeline. So an appropriate degree of parallelism is very important for a certain system. The second is the number of threads the system started. The thread of the system is directly related to the implementation efficiency. So, only appropriate amount of threads are started, the pipeline can have a good performance. The third one is the sub-problem size. The window size, or sub-problem size for each token, can also limit the throughput of the pipeline. Making windows too small may cause overheads to dominate the useful work. Making windows too large may cause them to spill out of cache. A good guideline is to try for a large window size that still fits in cache.

IV. EXPERIMENTAL DESIGN

In this paper, the pipeline classes offered by Intel Threading Building Blocks and C plus plus programming language will be used in the experiment of the remote sensing image processing. A complete image processing is divided into input, processing and output phases and each of the phases is relatively independent. A large image containing amount of remote sensing data is divided into many blocks of same size along horizontal and vertical directions. When the pipeline starts, the reading stage reads the blocks into memory in order serially, the processing stage processes the blocks in parallel, and the writing stage writes the processed blocks back into the file on the disks. This whole processing is like an assembly line without stopping, and this parallel approach can theoretically handle up to enhance efficiency. In the experiment, the size of the block, the number of started threads and the parallelism will affect the performance of the pipeline. The Lee filter is used to verify processing efficiency because the lee filter process is very complex and time-consuming. In the experiment, the lee filter will be done on a remote sensing image of 73M size in parallel based on pipeline, and the time spent on processing will be recorded as experimental data.

In Fig.5, it's shown that the pipeline have the best performance when the image is divided into blocks of 100*100 size and the number of the started threads is two, and the parallelism is three. The least time spent on processing in parallel is 7.438 seconds, and the processing of the same image in serial costs 8.0 seconds. So, the speedup is 1.08.

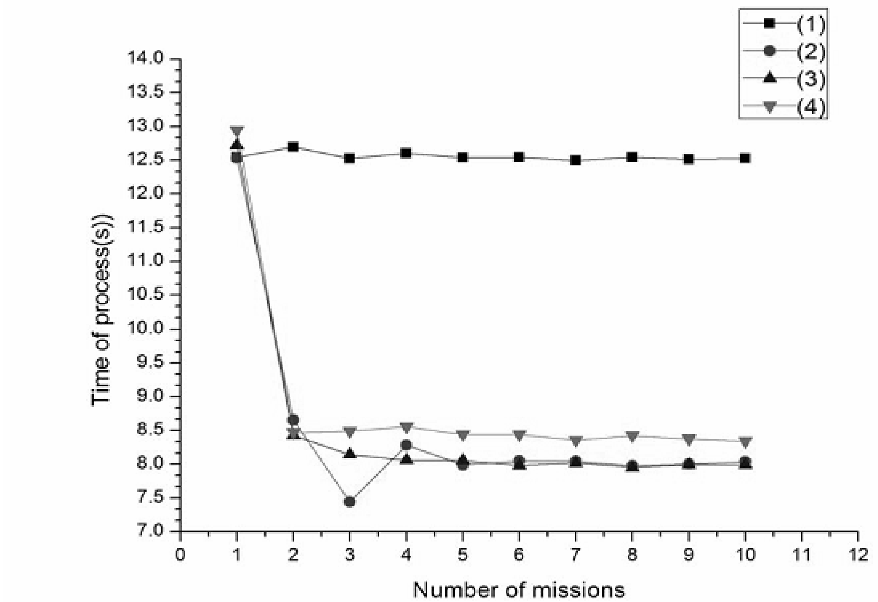

Figure 5. Time-missions-threads diagram

The Fig.6 shows that the pipeline has a better performance when the number of the started threads is two and the size of the sub-blocks is 400*400 pixels. And the time-parallelism curve is very smooth when the parallelism is more than two. The least time spent on processing is 5.808 seconds. The speedup is 1.3.

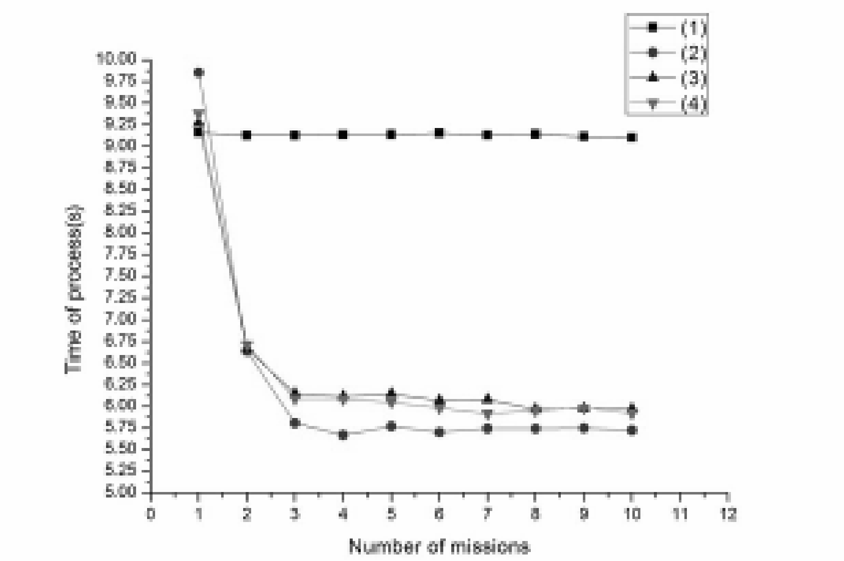

Figure 6. Time-missions threads diagram

The Fig.5 and Fig.6 also show that when the blocks size is small, the efficiency of the processing based on parallel is mainly affected by the I/O speed of the system. After a number of experiments, we find that the efficiency of the pipeline is higher than any other condition when the number of the started threads is two and the parallelism is four at the same time. Under this condition, we go further to study the relationship between the efficiency and the sub-block size. The obtained result is shown in Fig.7.

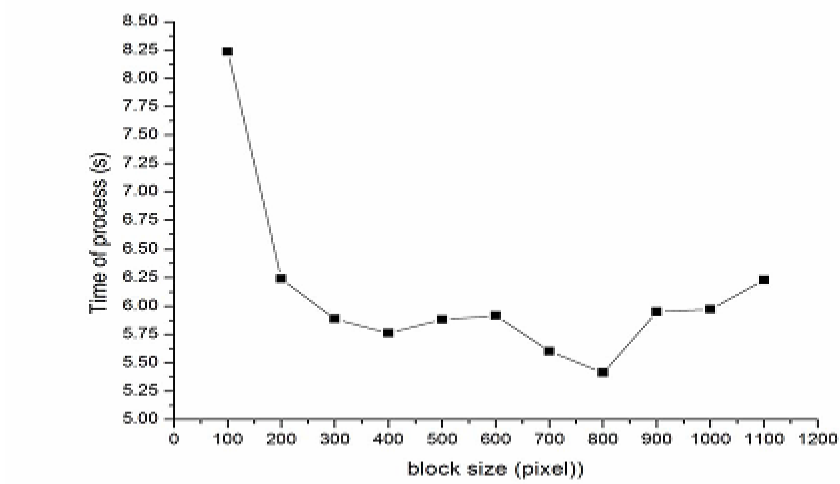

Figure 7. Time-size diagram

The Fig.7 shows that the block size has a great influence on the efficiency of the pipeline. The curve shows that the efficiency is highest when the number of the started threads is two and the parallelism is four and at the same time the size of the sub-blocks is 800*800 pixels. And the speedup is 1.43. The processing effect of image of larger amount of data based on pipeline is shown in Fig.8.

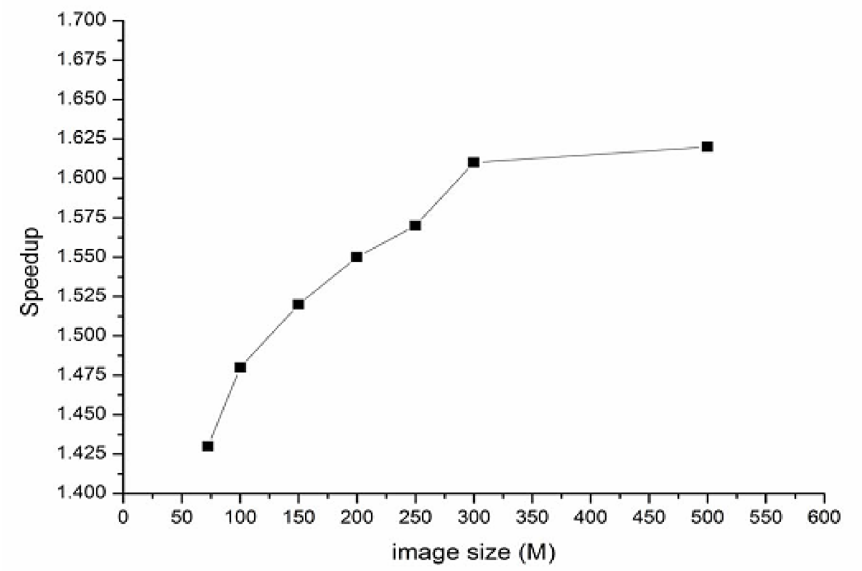

Figure 8. Speedup-size diagram

In the Fig.8, it is shown that with the amount of data increases, the speedup also increases. So, the pipeline will have a better performance in the processing of the image with larger amount of data.

V. Conclusion

The experiment shows that the pipeline model can significantly improve the speed of the processing of image with large amount of data. The pipeline is a feasible and effective solution for the efficient processing of remote sensing image. The pipeline can be used for automatic processing on a number of continuous processes and also can easily deal with non-parallel processes such as reading and writing. However, the pipeline technology is not suitable for processing the image of small amount of data because the efficiency of pipeline may be lower than that of serial execution for the overhead of the parallel system and interface of different processing blocks. How to improve the speed of processing the large remote sensing image is still a difficult issue and worth further study.

References

1. Zhifeng Xiao, Binglong Zhang, State Key Laboratory of Information Engineering in Surveying

2. Xingshu Hu, Jianya Gong, and Jianping Pan, "Current situation of remote sensing technology in the contemporary era and its developing tendency," Engineering Journal of Wuhan university, vol. 36(3A), pp. 195-198, 2003.

3. National Program for Medium-to-Long-Term Scientific and Technological Development (2006-2020).

4. Guorui Huang, Pin Zhang, and Guangbo Wei, "Key techniques of multi—core processor and its development trends," computer engineering and design, vol. 30(10), pp. 2414-2418, 2009.

5. Mattson T G, Sanders B A, and Massing ill B L, " Patterns for Parallel Programming," New Jersey: Prentice Hall, 2005.

6. Zhenhua Yang, Ming Chen, and Jianxun Zhao, "The Pipeline Technology in Computer," Science and Technology Information, pp. 79-80, 2009.

7. Chen G L, Sun G Z, and Xu Y, "Integrated research of parallel computing: Status and future," Chinese Sci Bull, vol. 54(11), pp. 1845¯1853, 2009.

8. Granma A, Gupta A, and Karypis G, "Introduction to parallel computing," Boston: Benjaming/ Cummings Publish Company, Inc., 2003

9. Intel® Threading Building Blocks Tutorial, pp. 27.

10. Yaming Li, "Computer organization and architecture," Tsinghua University Press, pp. 334-341, 2000(4).

11. Focai Peng, Cuihong Han, and Yongrui Zhang, Shuqun Shen, and Eryuan Zhao, "Research on Instruction Parallelism-based Software Pipeline," Microelectronics and Computer, vol. 20(12), pp. 1-3, 2003.