Multi-Grain Parallel Accelerate System for H.264 Encoder on ULTRASPARC T2

Yu Wang, Linda Wu, and Jing Guo
Key Lab. of the Academy of Equipment, Beijing, China
Email: {snailyu, guojing}@gmail.com, wulingda@163.com

***Abstract*—This paper describes a multi-grain parallel accelerate system for H.264 encoder on UltraSPARC T2 processor. This system integrates pipeline parallelism, frame-level, slice-level, macroblock-level data parallelism and SIMD technology.We use x264, an H.264 video encoder to implement our parallel accelerate system. Our implementation of parallel accelerate system achieves speedup between 10.1x and 11.5x and only causes 1.04x bit rate increase on UltraSPARC T2 processor.**

***Index Terms*—parallel video coding, H.264, SIMD, speedup**

I. Introduction

H.264 is the most recent and efficient standard for video compression [1]. Because it abstractly describes a wide range of compression tools and operating points .But this standard is so complex that some video encoders face difficulties encoding HD video sequences in real-time, even on recent processors. Many studies have endeavored to find ways to achieve high efficiency of video coding with higher speed and lower costs [2]. Among them, using parallel technology to obtain high computing power for video coding is a worthwhile method and has good application prospect.

At the same time, isomorphic multicore processors caused the concern of chip designers because of its simple structure, good scalability, compatibility, energy efficiency, fast and easy refactoring. Such as Intel and AMD are beginning to stop the single-core product development in favor of multi-core product development. This provides a good platform for the development of parallel programs [3]. Therefore many researchers have proposed many parallel strategies of H.264 decoding [4, 5] .But most of them are single-grained parallel. Some scholars have begun to study the multigrain parallel strategies of H.264 decoding[6].Based on the above researches, this paper presents a multi-granularity parallel strategy of H.264 decoding and implements the multi-granularity parallel decoder of H.264 on a SPARC T2 processor.

According to the size of the parallel granularity and parallel mode, H.264 encoding parallel categories can be divided into four groups: Coarse-grained control Parallelism, Fine-grained control Parallelism, Coarse-grained data parallelism, Fine-grained data parallelism just TABLE 1 shows. The coarse-grained data parallelism mainly refers to the data parallelism of macroblocks, slices, frames, or groups of pictures. The fine-grained data parallelism mainly refers to the SIMD technology which using the extension instruction set to optimize the algorithm[6].The coarse-grained control parallelism mainly refers to the pipeline processing of each module in H.264 encoding. There exists the following modules in H.264 encoding process: Intra-Prediction, Inter-Prediction, Motion Estimation, Motion Compensation, Discrete Cosine Transformation, Quantization, Reverse Quantization, Reverse DCT, Entropy encoding and Deblocking Filter. Coarse-grained control parallel takes advantage of code and data locality in H.264 processing, but it needs to equalize the arithmetic quantity of each module.

TABLE I.
PARALLEL CATEGORIES CLASSIFICATION OF H.264 ENCODING

Coarse-grained		Fine-grained
Data parallelism	Taking Macroblock, slice, frame as granularity	SIMD
Control parallelism	Pipeline parallelism	VLIW

The approaches show in Table I are not mutual. An excellent H.264 acceleration parallel system always integrate a variety of parallel methods. In this paper, we describe a novel parallel approach which accelerates pipeline parallelism, Frame-level, slice-level, macro block-level data parallelism and SIMD technology and takes advantage of the ULTRASPARC T2 processor architecture characteristics.

In this paper, we realize a multi-grain parallel H.264 encoder is implemented on the ULTRASPARC T2 multi-core platform. This encoder efficiently combines four parallel grains ,including frame ,slice, macroblock and data level. Experimental results demonstrate that the encoding speedup is improved to a large extent ,without obviously increasing bitrates. The high-quality video is kept in the encoding process.

The rest of the paper is organized as follows: Section II mainly describes the theoretical basis and the specific implementation method of the multi-grain parallel accelerate system for H.264 encode. Section III is the

Manuscript received October 9, 2012; revised March 12, 2013.

experimental comparison section. In this Section We do a multiple comparison tests to prove the superiority of the proposed method. Section IV makes a summary of this paper and briefly introduces our future work .

II. Parallel Acceleration System

Most H.264 video encoders consist of ten main functional units: Intra Analysis &Prediction Motion Estimation, Motion Compensation, Discrete Cosine Transformation, Quantization, Reverse Quantization, Reverse DCT, Entropy encoding and Deblocking Filter. Fig. 1 shows an execution profile for the serial X264 using manual code instrumentation, with one reference frame, 128×128 motion estimation search area, UMH search algorithm on SPARC. We observe that most of the execution time is spent in the analysis and encoding phases. According to the order of the decoding process and data of mutual dependency relationship, we divide the coding process into several modules for parallel design. Fig. 3 shows the threads assignment for our control parallel strategy.

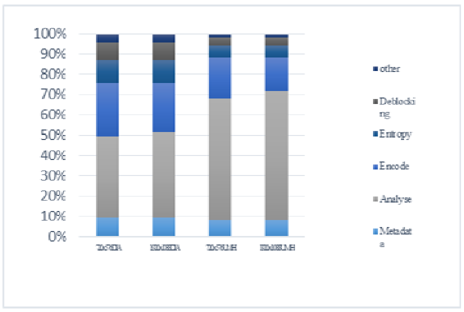

Figure 1. Normalized execution time of x264

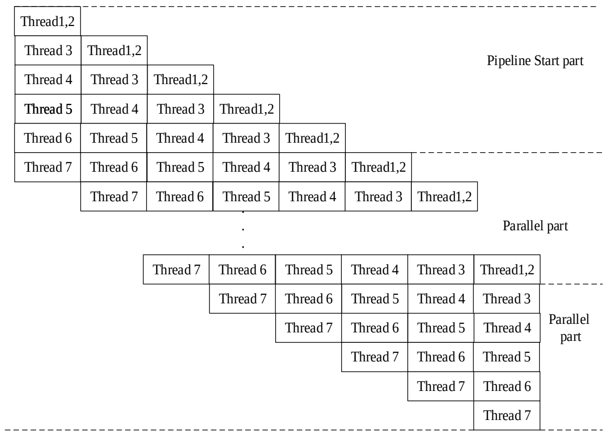
Thread1,2
Thread 3 Thread1,2
Thread 4 Thread 3 Thread1,2
Thread 5 Thread 4 Thread 3 Thread1,2
Thread 6 Thread 5 Thread 4 Thread 3 Thread1,2
Thread 7 Thread 6 Thread 5 Thread 4 Thread 3 Thread1,2
Thread 7 Thread 6 Thread 5 Thread 4 Thread 3 Thread1,2
Pipeline Start part
Parallel part
Thread 7 Thread 6 Thread 5 Thread 4 Thread 3 Thread1,2
Thread 7 Thread 6 Thread 5 Thread 4 Thread 3
Thread 7 Thread 6 Thread 5 Thread 4
Thread 7 Thread 6 Thread 5
Thread 7 Thread 6
Thread 7
Parallel part

Figure 2. Parallel pipeline.

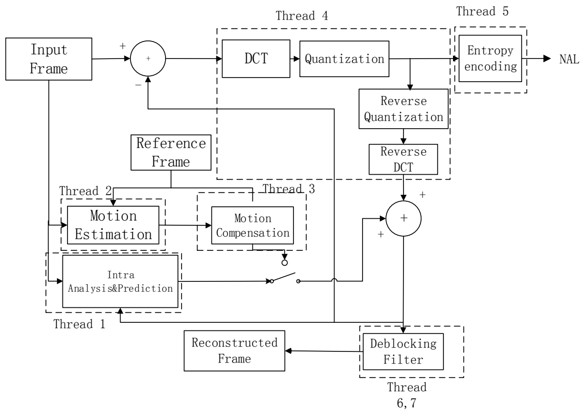

Figure 3. Threads assignment for the control parallel strategy

A. *Pipeline Parallel Strategy*

The entire pipeline consists three parts: pipeline startup, parallel and the pipeline end. Fig. 2 describes the specific arrangement of the pipeline. Just as the Fig. 2 shows, at the first cycle the seven threads start one by one when its data comes. When the first macroblock completes its coding, parallel processing begins. In order to improve processing speed, we allocate a memory for each thread. So that each thread can easily access the data to be processed. When the last macroblock begins its coding, the pipeline end part begins. The memory and the threads will be released step by step.

B. *Multi-frame Parallel Strategy*

In H.264 encoding I frame is internal coding frame and does not need to reference other frame. P frame needs forward I frame as a reference and B frame needs both forward and backward I frame or P frame as a reference. Due to the complex reference relationship between the frames, it is not simple to do frame coding parallel design. Therefore, finding the common reference frame is the key to realize the frame level parallel.

In the paper we propose a parallel strategy which can change with different B frame parameter. When the number of B frame in the coding sequence is changeable, the threads of frame parallel depends on the number of the B frame. The larger the B frame number is, the higher the speedup is. But creating threads and data transmission between the threads costs resources, the speedup will get its peak on a specific number of B frame.

C. *Multi-slice Parallel Strategy*

There is no direct data dependencies between the slices, so it is possible to encode multi-slices of the same frame. To realize the multi-slice parallel, each frame image will be divided into several average data blocks. Then we create a thread for every data block to process its data. Slice head will be added during the encoding process .When each thread completes encoding, we combine the slices into a frame image data. The degree of multi-slice parallelism depends on the slice number of a frame. Theoretically, we can divide a frame into as many slices as possible, so that we can get the highest speedup.

But when a frame is divided into several data blocks, the correlation between slices will cease to exist. This correlation can be well used in the macroblock prediction and can improve the compression ratio. So the disappearance of the correlation between slices will make compressed video`s bitrate increase. Therefore, it is not possible to divide frame into too many slices.

D. Data-based Parallel Strategy

SPARC multiple processors embeds the VIS multimedia instruction set, Using multimedia extensions instruction for optimization algorithm of large-scale intensive computing has obvious accelerating effect. In the X264 code, several calculation module is rewritten using multimedia extensions instruction. In detail it includes DCT, Motion Compensation and Deblocking Filter. In the X264 code, SIMD parallel has been realized using MMX and SSE. Using SIMD have evident accelerating effect on intensive computing. Fig. 5 shows the SIMD vector operation mode.

In our paper we realize the SIMD using VIS multimedia extensions INSTRUCTION. In order to increase accuracy, we use a flexible algorithm. FOR example we use the code as follow to calculate four 16-bit multiplication:

```
v4hi __builtin_vis_fmul8sux16 (v8qi, v4hi);
v4hi __builtin_vis_fmul8ulx16 (v8qi, v4hi);
v4hi __builtin_vis_fpadd16 (v4hi, v4hi);
```

We get a good parallel speedup by using Data-based Parallel Strategy. Besides Data-based Parallel Strategy used processor embedded multimedia instruction set which does not take up PROCESSOR resources. In Summary Data-based PARALLEL STRATEGY is a very effective parallel strategy in improving H.264 encoding speed and can save processor resources for Coarse-grained parallel strategy.

E. Realization of the H.264 Encoding Parallel Accelerate System

Fig. 4 shows the complete process of multi-grain parallel accelerate system. Based on the x264 open-source encoder, we create three data queues. The three queues are I/P coding frame queue, B frame coding queue and NAL output buffer queue. And we create classification threads to realize the three data queues. H.264 encoding process can be divided into three parts: original video input, encoding and coded data output .The involved threads are main thread, Read and write thread, I/P frame coding threads, B frame coding threads, slice-level threads and pipeline parallel threads. Read and write thread is responsible for reading image sequences from the buffer queue, determining frame type and inserting into the three frame coding queues; checking NAL queue and writing completed data back to the disk space. I/P frame coding thread is responsible for analyzing frame-level parallelism and creating B frame coding threads. The number of I/P frame threads and B frame threads is based on the number of slices. Inside each slices, create several threads to encode macroblocks with pipeline parallel strategy.

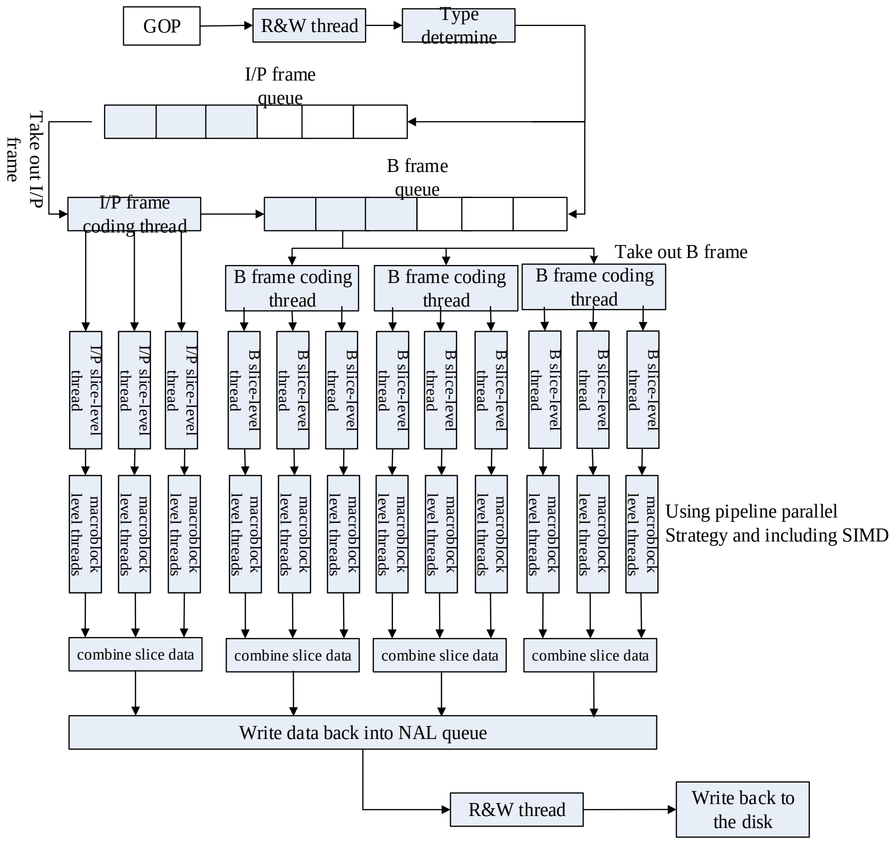

Figure 4. Compete process of multi-grain parallel accelerate system

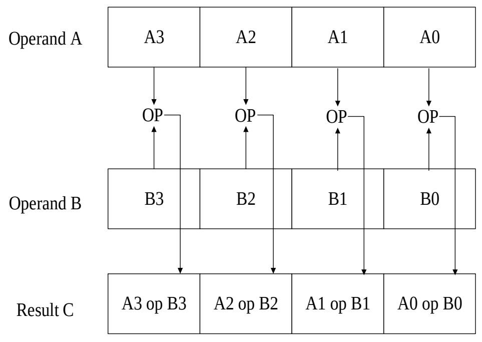

Figure 5. SIMD vector operation mod

Now, we will describe two major threads in the whole coding process by using pseudocode.

Read and Write thread code:

```
while(there is frame to encode)
{
        if(there is free entry in image buffer)
        {
        read a frame to the buffer;
        decide the type of current frame;
        if(the type is I or P)
                enter I / P queue;
        else
                enter B queue;
       allocate a node in the NAL queue for current frame;
        }
        else if(there is bitstream in the NAI.queue)
        write the hitstream to output file;
        else
        wait;
}
I/P frame thread code:
while(1)
{
        if(there is frame in the I / P queue)
        {
        fetch a frame from I / P queue;
        analyze the B frames in B queue;
        create B thread for B frame which can be encoded
        in parallel;
        encode current frame
        }
        else if(alI frames are encoded)
                exit;
        else
        wait;
}
```

III. Experimental Evaluation

Our experimental platform is ULTRASPARC T2 Multi-core processor with the Kylin Operating System developed by National University of Defense Technology of China. Kylin Operating System is developed based on Linux kernel. This paper mainly tests the speedup and bit rate changes of different granularity parallel strategy and the speedup changes of our acceleration system for different resolutions and different format video.

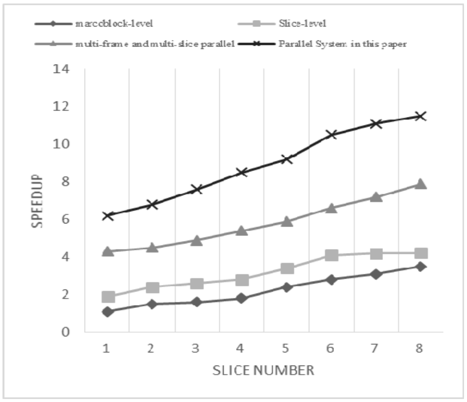

Figure 6. Speedup comparison of different granularity parallel strategy

A. *Different Granularity Parallel Strategy Speedup Comparison*

Fig. 6 shows four kinds of parallel granularity strategy speedup comparison. These parallel Granularity strategies are: Slice-level parallel, Macroblock-based parallel, Frame and Slice integration parallel and the parallel strategy in this paper. In Fig. 5 we can see that the speedup of four parallel strategies increases with the increase in the number of slices. The speedup of mcroblock-level parallel is about 3.5x while the slice-level is about 4.If we use Multi-Frame and Multi-Slice parallel, the speedup is about 7.6x, and the speedup in this paper is parallel strategies and ensure their effective and balanced running, we can get higher speedup.

As TABLE 2 shows, we compare the multi-granularity parallel strategy with other existing methods in speedup and bitrate. Chen [8] uses slice-level parallel to process video and achieves speedup between 4.1 and 4.23x.And this strategy causes 1.08x bit rate increase. Michail Alvanos[9] uses task-based parallel and achieves speedup between 4.7x and 8.6x. Stéphane Coulombe [6] describes a novel multi-frame and multi-slice parallel video encoding approach and achieves speedup of 7.2x. By the integration of multi-granularity parallel, our multi-granularity parallel encoding system achieves speedup between 10.1x and 11.5x and only causes 1.04x bit rate increase.

TABLE II.
PERFORMANCE COMPARISON OF THE PROPOSED METHOD AND OTHER METHODS

	Speedup	Bit-rate
Slice parallel	4.1x~4.23x	1.08x
Multi-frame and multi-slice parallel	7.2x~7.95x	1x
Task-based parallel	4.7x	1x
Parallel system in this paper	10.1x~11.5x	1.04x

B. *Experimental Results Analysis of Different Video Formats*

The video formats we use are: CIF(352×288),SIF(352×240), QCIF(176×144). As Fig. 7 shows, the higher speedup will be achieved if the video resolution is higher.

This is because the greater the resolution is, the computation time required in an image is also increasing. The cost of calculating is much more than the cost of the frame-level thread creation and the memory allocation in parallel encoding. Therefore, we can get a Conclusion that our parallel acceleration system will get more pronounced effect when processing high-resolution video.

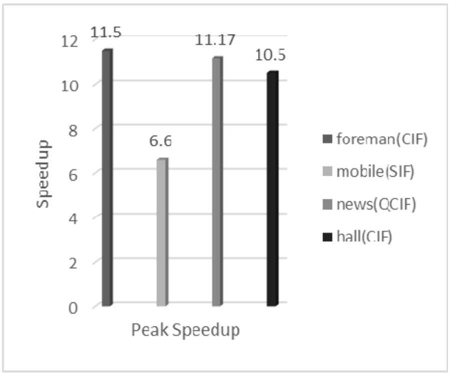

Figure 7. Multi-granularity parallel speedup comparison of different video format

IV. Conclusions

H.264 parallel encoding granularity contains coarse-grained control parallelism, fine-grained control parallelism, coarse-grained data parallelism, fine-grained data parallelism. The single granularity parallelization can increase speedup, but it usually gets low speedup and has the drawback of high bit. In this paper we describe a novel parallel approach which integrates pipeline parallelism, Frame-level, slice-level, macroblock-level data parallelism and SIMD technology and implement it on the SPARC T2 processor. Our parallel approach achieves a high speedup and gains superior encoding performance. Integrating different granularity parallel approaches is an effective way to gain high speedup and better encoding performance. The future, we will still study around this point.

References

[1] ISO/IEC 14496-10 and ITU-T Rec, “H.264, Advanced video coding for generic audiovisual services,” 2003.

[2] F. Niu, D. M. Li, and S. Peng, “Parallel coding efficiency analysis of H.264 on PC cluster,” *Symposium on Photonics and Optoelectronic*, Beijing, China, 2010, pp. 1-4.

[3] P. Jeff, D. John, and G. Bill, “From single core to multi-core: Preparing for a new exponential,” *IEEE International Conference on Computer-Aided Design*, New York, 2006, pp. 67-72.

[4] J. H. Ye and J. L. Liu, “Fast parallel implementation of H.264/AVC transform exploiting SIMD instructions,” *International Symposium on Intelligent Signal Processing and Communication Systems*, Xiamen, China, 2007, pp. 870-873.

[5] Z. Zhuo and L. Ping, “A highly efficient parallel algorithm for H.264 video encoder,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, 2006, pp. 489—492.

[6] J. F. Franche and S. Coulombe, “A multi-frame and multi-slice H.264 parallel video encoding approach with simultaneous encoding of prediction frames,” *Consumer Electronics, Communications and Networks*, 2012, pp. 3034-3038.

[7] T. Slingerland and J. Smith, “Measuring the performance of multimedia instruction sets,” *IEEE Transactions on Computers*, , vol. 51, no. 11, pp. 1317-1332, 2002.

[8] Y. K. Chen, X. Tian, and G. Steven, “Towards efficient multi-level threading of H.264 encoder on Intel hyper-threading architectures,” in *Proc. 18th Intel Parallel and Distributed Processing Symposium*, Santa Fe, 2004, pp. 63-72.

[9] M. Alvanos, G. Tzenakis, and S. Dimitrios, “Task-based parallel H.264 video encoding for explicit communication architectures,” *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, 2011, pp. 217-224.

Yu Wang was born in 1988, received Bachelor's degree in the Department of Computer Science and Technology in Tsinghua University in 2011. Now he is graduate student in the Academy of Equipment. He engages in the research of multi-media technology

Lingda Wu was born in 1962. She is a professor at Key Lab. of the Academy of Equipment. Her research interests are mainly in virtual reality technology and multi-media technology.

Jing Guo was born in1977, received Ph.D. degree from National University of Defense Technology in 2008. Now she is an assistant researcher in the Academy of Equipment. She engages in the research of system science.