Hierarchical Architecture of JPEG2000 Parallel Encoder on Multi-GPU Cluster System

Bumho Kim, Jeong-Woo Lee, and Ki-Song Yoon

***Abstract*—There has been an increase in the demand for a high-quality video codec that supports 4K (3,840×2,160) or more. JPEG2000 is an important technique for data compression, which has been successfully used in digital cinema and medical application. To process the high workload of JPEG2000 coding for large-scale video data, hybrid CPU/GPU platform is used to obtain high computing power. This paper describes the implementation of the JPEG2000 compression standard in multi-GPU platforms. Specifically, we propose the scalable cluster architecture of the JPEG2000 encoder to achieve scalability and high encoding speed. The proposed scheme can reduce the large encoding time and significantly improve the coding efficiency. The proposed scalable cluster system is very suitable for high-resolution video such as 4K or 8K containing large amounts of video data.**

***Index Terms*—JPEG2000, parallel system, hybrid platform, GPGPU, digital cinema.**

I. Introduction

There has been an increase in the demand for a high-quality video codec that supports a resolution of 4K (3,840×2,160) or 8K (7,680×4,320) pixels. JPEG2000 compression standard is an important technique for image compression [1], which has been successfully used in both mobile applications as well as high-quality applications such as medical imaging and digital cinema. Due to the increasing spatial resolution of digital cinema and medical images, fast compression of image data is becoming an important and challenging objective.

The JPEG2000 codec, which shows a higher compression and enables higher resolutions, entails a much more complex process with an enormous amount of computations. For 4K and 8K video containing large amounts of data, JPEG2000 encoders have a very high CPU demand, and it is hard for a single core computer to deal with such complex coding computations [2], [3].

To process the high workload of JPEG2000 coding for large-scale video data, hybrid CPU/GPU platform is used to obtain high computing power. During recent years, multimedia software has been ported to multi-core and GPU architecture. Transition to hybrid CPU/GPU platforms in high performance computing is challenging in the aspect of efficient utilization of the heterogeneous hardware and existing optimized software [4].

This paper describes the implementation of the JPEG2000 compression standard in hybrid CPU/GPU platforms to obtain high computing power and balance the load between cores and GPUs in the hybrid architecture. Specifically, we propose a scalable cluster architecture of the JPEG2000 encoder on multi-GPU system. The cluster architecture is a distributed system that consists of interconnected computers working together as an integrated unit. The simulation results verify that the proposed scalable cluster system is very suitable for highly complex video coding that involves a large amount of computations.

The rest of the paper is organized as follows. Section II and Section III address JPEG2000 and GPGPU respectively. The proposed scheme and system model are presented in Section IV. The performance results of the proposed method are shown in Section V. Finally, Section VI concludes remarks on the proposed scheme.

II. JPEG2000 Overview

JPEG2000 is an image compression standard from Joint Photographic Experts Group (JPEG). JPEG2000 provides compression performance superior to the current standards but also advanced features demanded by today's emerging applications. JPEG2000 standard is based on wavelet technology and a layered file format that offer flexible lossy-to-lossless compression, irreversible compression that preserve image accuracy, and advanced functionality of image data management systems. The JPEG2000 also provides great scalability in both quality and resolution and can work in both lossy and lossless mode on very large images.

To meet these needs, JPEG2000 adopts a number of contemporary digital signal processing methods including a discrete wavelet transform (DWT) and embedded block coding with optimized truncation (EBCOT). The process DWT is a sub-band transform which transforms images from the spatial domain to frequency domain. Therefore, DWT can efficiently exploit the spatial correlation between pixels in an image. EBCOT is a two-tiered coder: Tier-1 is responsible for bit plane coding (BPC) and context adaptive arithmetic encoding (AE); Tier-2 handles rate-distortion optimization and bitstream layer formation. Fig. 1 shows simplified block diagram of compression system defined by JPEG2000 standard.

These advanced features and the superior compression performance yields higher computational demands which implies slower processing. Slow performance has long been noted as a major drawback of JPEG2000, particularly in software implementations. Resulting computational

Manuscript received December 5, 2014; revised January 19, 2015. This work was supported by the Ministry of Science, ICT and Future Planning (14-000-02-001 Development of UHD Realistic Broadcasting, Digital Cinema, and Digital Signage Convergence Service Technology).

The authors are with the ETRI (Electronics and Telecommunications Research Institute), Daejeon, South Korea (e-mail: mots@etri.re.kr, jeongwoo@etri.re.kr, ksyoon@etri.re.kr).

DOI: 10.7763/IJIEE.2015.V5.565

requirements of JPEG2000 are one of drawbacks hindering use of JPEG2000 in common application.

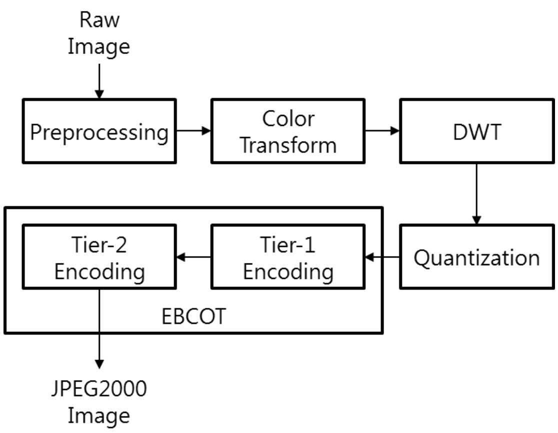

Fig. 1. JPEG2000 encoding process.

III. GPU Overview

Graphics processing units (GPUs) have become a popular computing architecture in recent years due to rapid increase of performance as compared to traditional CPUs [5].

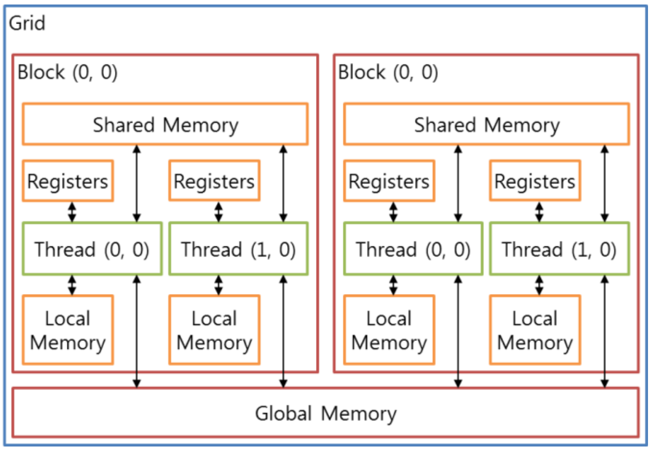

Fig. 2. GPU architecture.

CUDA is software and hardware platform designed for general purpose computing on GPUs in order to take full advantage of the maximum performance of GPUs in applications [6]. GPUs have a parallel architecture capable of running thousands of threads in parallel shown in Fig. 2.

In CUDA computing model, threads are grouped into thread blocks, and threads within thread block can cooperate among themselves using synchronization primitives by sharing data via a global memory and shared memory. The advantage of the global memory is that it can be accessed by all threads directly, whereas the shared memory is only accessible to threads of one block. Compared to the global memory, the shared memory is considerably smaller and significantly faster. The data can be partitioned and fetched into the shared memory to provide higher throughput for more complex operations. While these architecture specifics of GPUs allow fine-grained parallelization for impressive increase of performance, it requires adaptation and re-formulation of algorithms resulting in more effective design on the GPU.

IV. System Model

There has been a lot of effort to provide JPEG2000 applications with sufficient processing speed [7], [8]. This paper proposes a parallel architecture of the JPEG2000 encoder in hybrid CPU/GPU platform to achieve scalability and a high encoding speed. To process the high workload of JPEG2000 encoding for large-scale video data, we develop the implementation of parallel encoder using multi-core CPU and multi GPUs [9].

Fig. 3 shows a simplified block diagram of the JPEG2000 encoder to enhance the coding performance using multicore CPU and multi-GPUs platform. As mentioned in Section II, the JPEG2000 encoder consists of several steps that are performed in consecutive order.

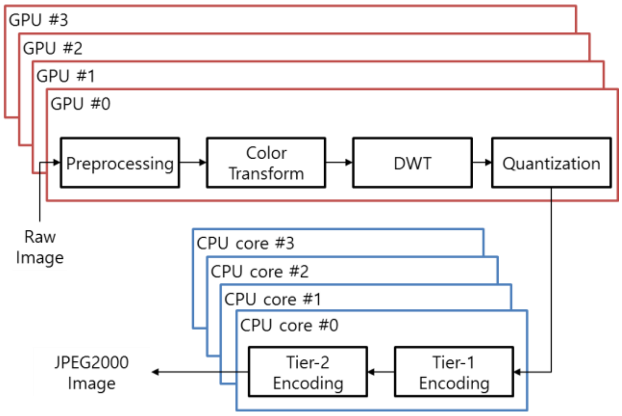

Fig. 3. JPEG2000 encoder architecture on multi-core CPU and multi-GPU system.

The first encoding step is component transform which converts the multiple color components data into another color representation. The component transform removes the inter-component redundancy that could be found in the image.

The next step is DWT which is a domain transform that transforms an image from special domain to frequency domain. This enables an intra-component special decorrelation that concentrates the image information in a small localized area.

Once DWT is applied, all the resulting wavelet information is quantized, which means that wavelet coefficients are reduced in precision. The process of quantization introduces reduction of the data precision in order to achieve compression.

The encoding processes up to quantization are performed in multiple GPUs. In order to obtain high efficiency, the each component could be processed independently on separate GPUs. The first work flow is to copy image data from CPU RAM to global memory of GPU. Once image data is ready in global memory, the encoding process from color transform to quantization can be executed on GPUs.

After quantization, the integer wavelet coefficients still contain a lot of spatial redundancy. This redundancy is removed by context-based entropy coding (EBCOT) Tier-1 so the data is efficiently compressed into a minimum size bit-stream. EBCOT Tier-2 process is creating and ordering the packets for rate allocation. These processes of entropy coding is highly sequential and difficult to parallelize efficiently using many threads in GPU. Therefore, EBCOT step is performed in CPU. Each of these code-blocks is entropy coded separately, which gives potential for parallelization in multi-core CPU. At the end of the computations all the data have to be saved on the CPU memory.

Fig. 4 shows the hierarchical architecture of the JPEG2000 encoder used to enhance the coding performance using a cluster platform. This cluster platform is a parallel and distributed type system, which consists of a collection of connected computers working together [10]. Each node can be a single or multi-core system, and is connected through the network. The cluster platform can cope with large-scale applications with high performance [11], [12].

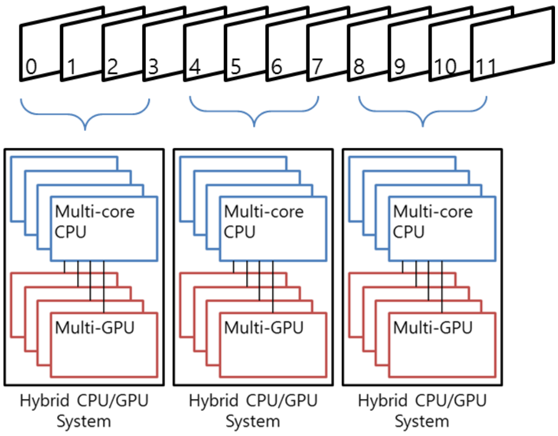

Fig. 4. Hierarchical architecture of the JPEG2000 encoder.

In the first level, the master node divides the input image sequence into a block of frames. Every block of frames is sent to each node and assigned to processes inside the node.

Each node encodes block of frames independently of the other nodes using the JPEG2000 parallelism method. When one block of frames is encoded completely, the coded bitstream is sent to the master node, and each node encodes the next block of frames.

V. Simulation Results

To measure the performance of the proposed approach, the notations shown in Table I are used.

TABLE I: NOTATIONS

Options	Value
Ts	Encoding time of each frame in sequential
Tp	Encoding time of each frame in parallel
Fs	Size of each frame
Nb	Network bandwidth
Nt	Network transfer time of each frame
N	Number of nodes

Let T_s and T_p denote the encoding time of each frame sequentially and in parallel, respectively. The efficiency is calculated by [13].

$$\text{Effciency}=\frac{T_p}{T_s \times N}. \tag{1}$$

If we consider the network transfer time of each frame, the encoding time in parallel should be

$$T_p = E_p + N_t \,. \tag{2}$$

The network transfer time (N_t) of each frame is calculate by

$$N_t = \frac{F_s}{N_b} \,. \tag{3}$$

In the cluster platform, the N_{th} node should wait for one block of frames until

$$N_w = (N-1) \times N_t \,. \tag{4}$$

If it is assumed that each node can be received in a block of frames when encoding the previous block of frames, no idle time exists. Therefore, the efficiency should be

$$Efficiency = \frac{E_p + N_t}{E_s \times N} \,. \tag{5}$$

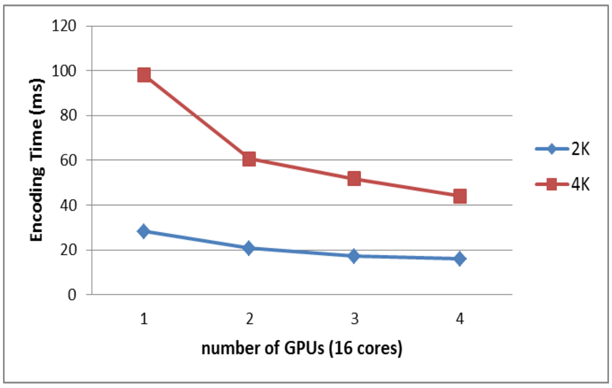

Fig. 5. Encoding time of the proposed scheme (16 cores CPUs).

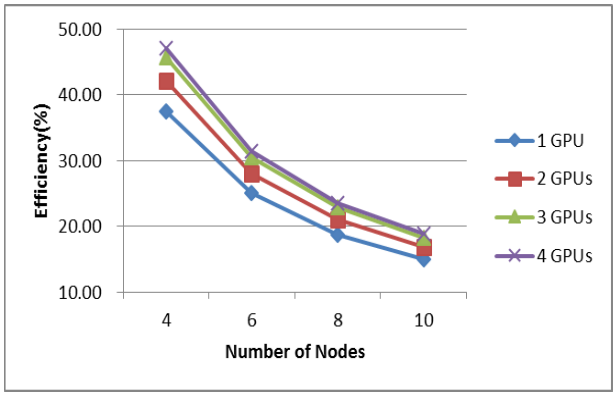

Fig. 6. Efficiency of the proposed scheme.

The proposed JPEG2000 encoder are implemented in the reference software, called "JasPer" [14], which is defined in Part 5 of the JPEG2000 standard. The Jasper encoder has been profiled to evaluate the proposed JPEG2000 encoder which is modified to be in parallel. The dual CPUs with an Intel Xeon w5590 at 3.33GHz clock frequency and NVidia Geforce GTX 680 GPUs with CUDA are used in this experiment. Four sets of test sequences were selected, i.e., 2K (1920×1080, 2160×1080) and 4K (3840×2160, 4096×2160).

Fig. 5 shows the total encoding time of multi-GPU parallel processing for each resolution. This experiment shows the efficiency of the proposed multi-GPU JPEG2000 encoder, which use 16 cores CPU.

VI. Conclusion

This paper proposed a scalable cluster architecture of the JPEG2000 encoder by combining two levels of parallelism, multi frame level parallelism and the hybrid CPU/GPU parallel method, to achieve scalability and a high encoding speed To process the high workload of JPEG2000 encoding for large-scale video data, this paper developed the implementation of parallel encoder using multi-core CPU and multi GPUs. To implement a parallel encoder the additional data parallelism, the multi frame level partitioning is adopted.

The simulation results verify that the proposed scalable cluster system is very suitable for highly complex video coding that involves a large amount of computations. Particularly for high-resolution video, i.e., 4K and 8K video that contain large amounts of video data, the proposed scheme can reduce the large encoding time and significantly improve the coding efficiency. The proposed architecture can be applied to implement a JPEG2000 encoder for high-resolution video sequences. As future work, we plan to implement an additional parallelism level by introducing SIMD and OpenMP to achieve real-time encoding.

References

[1] JPEG2000 image coding system — Part 1: Core coding system, *Information Technology*, ISO/IEC 15 444-1, Dec. 2000.

[2] M. Ciznicki, K. Kurowski, and A. Plaza, "Graphics processing unit implementation of JPEG2000 for hyperspectral image compression," *Journal of Applied Remote Sensing,* vol. 6, June 2012.

[3] Le, Roto, I. R. Bahar, and J. L. Mundy, "A novel parallel Tier-1 coder for JPEG2000 using GPUs," in *Proc. 2011 IEEE 9th Symposium on Application Specific Processors (SASP),* IEEE, pp. 129-136, 2011.

[4] R. Le, J. L. Mundy, and R. Bahar, "High performance parallel JPEG2000 streaming decoder using GPGPU-CPU heterogeneous system," in *Proc. 2012 IEEE 23rd International Conference on Application-Specific Systems, Architectures and Processors (ASAP),* IEEE, pp. 16-23, 2012.

[5] C. Miłosz *et al.*, "Benchmarking JPEG 2000 implementations on modern CPU and GPU architectures," *Journal of Computational Science*, vol. 5, no. 2, pp. 90-98, 2013.

[6] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley, 2011.

[7] J. Matela, V. Rusnak, and P. Holub, "GPU-based sample-parallel context modeling for EBCOT in JPEG2000," *OASIcs-Open Access Series in Informatics,* vol. 16, 2010.

[8] J. Matela, "GPU-Based DWT acceleration for JPEG2000," in *Proc. Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pp. 136-143, 2009.

[9] B. Kim, J. Lee, and K. Yoon, "A parallel implementation of JPEG2000 encoder on Multi-GPU system," in *Proc. 2014 16th International Conference on Advanced Communication Technology (ICACT),* IEEE, pp. 610-613, 2014.

[10] S. Sharifian, S. A. Motamedi, and M. K. Akbari, "Estimation-based load-balancing with admission control for cluster web servers," *ETRI Journal*, vol. 31, no. 2, pp. 173-181, Apr. 2009.

[11] A. Rodriguez, A. González, and M. P. Malumbres, "Performance evaluation of parallel MPEG-4 video coding algorithms on clusters of workstations," in *Proc. IEEE Int. Conference on Parallel Computing in Electrical Engineering*, IEEE, pp. 354-357, 2004.

[12] B. Kim, J. Lee, Y. Jeong, and K. Yoon, "Hierarchical architecture for HEVC parallel encoder," *Journal of Advanced Information Technology and Convergence,* vol. 3, no. 2, pp. 41-51, Dec. 2013.

[13] F. Niu, D. Li, and S. Peng, "Parallel coding efficiency analysis of H. 264 on PC cluster," in *Proc. Symposium on Photonics and Optoelectronic (SOPO)*, IEEE, pp. 1-4, 2010.

[14] M. D. Adams and F. Kossentini, "JasPer: A software-based JPEG-2000 codec implementation," in *Proc. IEEE Int. Conf. Image Processing,* vol. 2, pp. 53-56, Oct. 2000.

Bumho Kim received the BS degree in computer science from Sogang University in 2000 and MS degree in information technology from Information Communication University in 2002, respectively. Currently, he is a senior researcher in the Creative Content Research Lab. at Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. His research interests include multimedia, video codec, digital cinema, and digital contents distribution.

Jeong-Woo Lee received the B.S. degree in information and telecommunication engineering from Jeonbuk National University, Jeonju, Korea, in 1996, and the M.S. degree in information and communications engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 1998. He received the Ph.D. degree in the Information and Communications Department from GIST in 2003. He is currently working in Electronics and Telecommunications Research Institute (ETRI). His research interests include digital video coding algorithms, implementations for H.264 and HEVC, rate control algorithms for video coding, scalable video compression, and GPU-based coding algorithms.

Ki-Song Yoon received his M.S. and Ph.D degrees in computer science from New York City University in 1988 and 1993 respectively. From 1993, he was a principal member of Electronics and Telecommunications Research Institute (ETRI). His research interests are digital contents distribution, digital rights management and digital cinema/signage.