Published in final edited form as:

Proc SPIE Int Soc Opt Eng. 2014 March 19; 9039: 90390U-. doi:10.1117/12.2043869.

Analysis of scalability of high-performance 3D image processing platform for virtual colonoscopy

Hiroyuki Yoshida*, Yin Wu, and Wenli Cai

3D Imaging Research, Department of Radiology, Massachusetts General Hospital and Harvard Medical School, 25 New Chardon St., Suite 400C, Boston, MA 02114

Abstract

One of the key challenges in three-dimensional (3D) medical imaging is to enable the fast turn-around time, which is often required for interactive or real-time response. This inevitably requires not only high computational power but also high memory bandwidth due to the massive amount of data that need to be processed. For this purpose, we previously developed a software platform for high-performance 3D medical image processing, called HPC 3D-MIP platform, which employs increasingly available and affordable commodity computing systems such as the multicore, cluster, and cloud computing systems. To achieve scalable high-performance computing, the platform employed size-adaptive, distributable block volumes as a core data structure for efficient parallelization of a wide range of 3D-MIP algorithms, supported task scheduling for efficient load distribution and balancing, and consisted of a layered parallel software libraries that allow image processing applications to share the common functionalities. We evaluated the performance of the HPC 3D-MIP platform by applying it to computationally intensive processes in virtual colonoscopy. Experimental results showed a 12-fold performance improvement on a workstation with 12-core CPUs over the original sequential implementation of the processes, indicating the efficiency of the platform. Analysis of performance scalability based on the Amdahl's law for symmetric multicore chips showed the potential of a high performance scalability of the HPC 3D-MIP platform when a larger number of cores is available.

Keywords

3D imaging processing; high-performance computing; performance scalability; virtual colonoscopy

1. INTRODUCTION

In clinical practice, there is an increasing demand for fast-turnaround time in obtaining high quality, multi-dimensional images directly from picture archiving and communications system (PACS). In particular, real-time 3D imaging and post processing is an area with promising potential for a wide range of applications. However, high-speed 3D medical image processing (3D-MIP) is challenging because of the requirement of both high computational power and data throughput. In addition, the 3D imaging algorithms are time-

*yoshida.hiro@mgh.harvard.edu; phone 1 617 643-2326; fax 1 617 643-2743; www.massgeneral3dimaging.org.

consuming due to the large amount of data (several gigabytes) which often needed to be processed with iterative algorithms.

Although several hardware-based approaches have been made to achieve the high performance, they are not applicable to the wide range of 3D-MIP applications required in clinical practice. On the contrary, increasingly available and affordable high-performance multi-processor systems such as multicore and many core systems, and cluster and cloud computing systems provide a promising avenue to address the above challenges and have the potential to be adaptive to the 3D-MIP systems in clinical practice.

We previously developed a software platform that is designed to support scalable, high-performance 3D medical image processing, called HPC 3D-MIP platform, using increasingly affordable commodity computing systems¹. In this study, we evaluated the performance of the HPC 3D-MIP platform by applying it to computationally intensive processes in virtual colonoscopy, and analyzed the scalability of the performance based on the Amdahl's law for symmetric multicore chips².

2. MATERIALS AND METHODS

2.1 Scalable HPC 3D-MIP computing platform

This section briefly describes the HPC 3D-MIP system that we have developed previously¹.

2.1.1 Distributed block volume structure—The HPC 3D-MIP platform employs distributed block volume structures (DBVS) as the fundamental data structure to enable efficient parallelization and optimization of a wide range of 3D-MIP algorithms. The volume data are divided into non-overlapping set of blocks for parallel image processing. The DBVS provides the following advantages: (a) It segregates background voxels into a shared background block for obviating unnecessary computation and data communication (Figure 1 right); (b) a block naturally serves as a data unit to be distributed for parallel processing; and (c) the DBVS supports an adaptive block size to optimize for parallelism and performance.

2.1.2 Wave-front parallel task scheduling scheme—The wave-front parallelism task scheduling method is used on volume block tiles to load-balance tasks among computing units in order to minimize data wait and lock as well as to maximize local data reuse³. Many of the 3D-MIP algorithms contain nests of loops where the same data are reused in successive iterations of the outermost loop. In addition, many of the 3D-MIP algorithm process voxels that have dependence only on the voxels within the block and possibly neighboring blocks. These voxels can be achieved by processing various portions of the blocks, block slabs, or block tiles in parallel.

2.1.3 Layered parallel 3D image processing libraries—The software layers in HPC-3D are designed for easy parallelization of a wide range of 3D-MIP algorithms (Figure 1 left). The bottom layer abstracts the hardware platform to enable targeting of any multicore CPU-based computing system, cluster of computers, or cloud computing system. The rest of the layers are 3D-MIP software layers that provide the fundamental mechanisms

for high-performance parallel processing. Parallelization of the image processing library uses Microsoft Parallel Patterns Library (PPL). We added a special speed-up function in our parallelization for the commonly used, computationally expensive algorithms, which was created on top of the Intel Integrated Performance Primitives (IPP).

2.2 Application to computationally intensive modules in virtual colonoscopy

Virtual colonoscopy, also known as CT colonography, is a viable alternative to optical colonoscopy for diagnosis of colorectal cancers^{4, 5}. Electronic cleansing (EC) is one of the most computationally intensive processes in virtual colonoscopy, in which contrast-enhanced fecal materials in the colon are virtually cleansed to reveal colonic lesions submerged in the feces⁶. We evaluated the performance of HPC 3D-MIP platform by applying it to our previously developed EC system in virtual colonoscopy⁶ to develop a high-performance electronic cleansing (HPC-EC) scheme. To implement the HPC-EC, the following modules were developed on top of the HPC 3D-MIP platform¹: (1) Parallel segmentation of the colon by use of parallel region growing method; (2) parallel structure analysis of the colonic lumen by use of the parallel Gaussian and Hessian filters for computing local morphologic features; (3) parallel local roughness analysis by use of parallel Gaussian derivative filters; (4) parallel dynamic-threshold level set method by applying the wave-front parallel scheduling (section 2.1.2) to the dynamic-threshold level set method⁷, which was used to segment and remove the tagged fecal materials to recover the colonic wall while preserving submerged soft-tissue structures.

2.3 Performance scalability model

Gene Amdahl⁸ proposed a simple yet generic performance scalability model for a software system, in which n processors can be used in parallel for achieving high-performance computing. His performance scalability model, which is known as Amdahl's law to date, assumes that f is the proportion of a program that was in nitely parallelizable with no overhead for scheduling, communication, and synchronization, while the remaining fraction, $1-f$, remained completely sequential. By use of n processors in parallel, f portion of the program becomes n times faster, whereas $1-f$ portion of the program remains as is. Thus, the maximum speedup that can be achieved by using n parallel processes is

$$\text{Speedup}(f, n) = \frac{1}{(1-f) + \frac{f}{n}} \quad (1)$$

Despite its simplicity, Amdahl's law applied broadly and gave important insights such as the following: In order to benefit from a large number of processors, the parallelization fraction f must be large, and even if n approaches in nity, speedup is bounded by $\frac{1}{1-f}$, as only the second term in the denominator in the above equation becomes zero. The typical value of f has been large enough to favor a single processor. Thus, mainframes with one or a few processors dominated the computing landscape, and this trend was largely held in the minicomputer and personal computer eras. Even in the multicore or many-core eras, Amdahl's law still holds for performance scalability.

Hill and Marty² extended the Amdahl's law to multicore processors by regarding the number and performance of cores that a processor can support as adjustable parameters. The model, which is known as Hill-Marty model, assumes that a multicore processor can contain n base core equivalents (BCE), in which a single BCE implements the baseline core.

The Hill-Marty model also assumes that multiple BCEs can be combined to generate a core with greater sequential performance. Let the performance of a single-BCE core be 1, and let $\text{Perf}(r)$ be the sequential performance of a core that is created by using r BCE resources. In a typical case, $1 < \text{Perf}(r) < r$; thus increasing core performance increases sequential execution but decrease the performance of parallel execution.

Hill and Marty classified the architecture of multicore chips into three types²: symmetric, asymmetric, and dynamic multicore processors. A symmetric multicore processor requires that all its cores have the same cost. A symmetric multicore processor with a resource of n BCEs has $c = n/r$ cores of r BCEs each. Based on Amdahl's law, the speedup of a symmetric multicore processor relative to using one single-BCE core depends on the program's parallelization fraction f , the processor resources in BCEs n , and the BCE resources r that are devoted to increase each cores performance. The processor uses one core to execute sequentially at performance $\text{Perf}(r)$. Thus, the Hill-Marty model is represented by

$$\text{Speedup}(f, n, r) = \frac{1}{\frac{1-f}{\text{Perf}(f)} + \frac{f \cdot r}{\text{Perf}(r) \cdot n}}. \quad (2)$$

Let c be the number of cores executed in parallel at performance $\text{Perf}(r)$, i.e., $c = n/r$.

Following Hill and Marty², let us assume that $\text{Perf}(r) = \sqrt{r}$, although $\text{Perf}(r)$ can be any arbitrary function of r . Then, the Hill-Martin's model can be represented by

$$\text{Speedup}(f, n, c) = \frac{1}{\frac{1-f}{\text{Perf}(n/c)} + \frac{f}{\text{Perf}(n/c) \cdot c}} = \frac{1}{\frac{1-f}{\sqrt{n/c}} + \frac{f}{\sqrt{n/c} \cdot c}}. \quad (3)$$

For fixed n and c , $\text{Speedup}(f, n, c)$ is an increasing function of f . Thus, Amdahl's law applies to multicore processors because achieving the best speedups requires f being near 1.0, i.e., almost perfect parallelization. For fixed f and n , however, there is a tradeoff, and the maximum speedup can occur at a certain number of cores, and the speedup may decrease after this number. The Hill-Marty model provides a simple yet generic performance scalability model for a software system that runs on a contemporary multicore processor hardware system.

In this study, the Hill-Marty model is used as a performance scalability model for the HPC 3D-MIP platform. The model is fit to the empirical speedup data obtained from the HPC-EC scheme running on the platform, and the resulting performance curve is extrapolated with respect to the number of cores to estimate the performance scalability of the platform. The model fitting to the empirical data and extrapolation of the fitted curve were performed by use of Mathematica (Wolfram Research, Champaign, IL, USA).

3. RESULTS

The evaluation of the execution speed of the HPC-EC scheme was performed based on computational workstations equipped with 2-, 4-, 8-, and 12-core CPUs (Intel Xeon, 3 GHz) and 8 GB memory, and compared against the speed of the original, sequential EC scheme. Table 1 shows the average execution time of the four modules of HPC-EC described in Section 2.2 across 50 sets of virtual colonoscopy image data. All images in the dataset had x , y dimension of 512×512 pixels, with varying z dimension ranging from 340 to 600 pixels.

As shown in the table, each of the modules in HPC-EC was sped up when the number of cores was increased. The execution time for colon segmentation module was reduced from 5 min to 9 sec on the sequential EC and the HPC-EC, respectively, yielding 33-fold speedup in computation time on the HPC 3D-MIP platform; structure analysis module reduced from 4 min to 21 sec, yielding 12-fold speedup; roughness analysis module reduced from 7 min to 35 sec, yielding 12-fold speedup; and dynamic level set method module reduced from 12 min to 90 sec, yielding 8-fold speedup. The total execution time was reduced from 28 min to 2.3 min, yielding a 12-fold speedup. These results indicate that the HPC 3D-MIP platform is effective in enabling high-performance processing of the EC modules.

Figure 2 shows that the individual module in the EC were also sped up at a similar or greater proportion, and that the speedup of the total and individual processes in HPC-EC increases as the number of CPU cores increase, indicating the effect of excellent parallelization in the HPC-EC modules. This figure also shows that the speedup was not a linear function of the number of cores, as expected from the Amdahl's law².

Figure 3(a) shows the graph of the Hill-Martin model², (f, n, c) in Eq. (3), being fit to the speedup data of the entire HPC-EC process (red circles) by regarding parallelization fraction, f , and processor resources in BCEs, n , as fitting parameters. The estimated parallelization fraction was $f = 0.99$ at $n = 14$, indicating that the HPC-EC was highly parallelized.

To estimate the scalability of the HPC-EC when a larger number of cores is available, we extrapolated the fitted performance curve by increasing the number of cores beyond n (i.e., $c, > n$) in the Hill-Martin model. This is essentially equivalent to assuming that a BCE can be further divided into a multiple small BCEs without reducing the performance of the divided BCEs. Figure 3(b) shows the performance scalability curve obtained by extrapolating the Hill-Marty model up to 128 cores. The extrapolated performance curve shows the potential of a high scalability of the HPC-EC when a larger number of cores is available, although the maximum speedup is limited to 24 times that of the single core, when the number of cores reaches 128.

4. CONCLUSIONS

We have developed a software platform, called the HPC 3D-MIP platform, which was designed to support high-performance 3D-MIP using commodity parallel computing systems. Performance analysis of the HPC 3D-MIP platform has shown that the platform was effective in substantially speeding up computationally intensive processes in virtual

colonoscopy. Scalability analysis has shown that the platform has the high potential of performance scalability when a larger number of cores are available. Together, the HPC 3D-MIP platform is promising in providing a high-performance highly scalable computational platform for 3D medical image processing.

Acknowledgments

The project described was supported in part by grant R01CA131718 and R01CA166816 from the National Cancer Institute at the National Institutes of Health.

References

1. Yoshida H, Wu Y, Cai W, Brett B. Scalable, high-performance 3D image computing platform: system architecture and application to virtual colonoscopy. *Conf Proc IEEE Eng Med Biol Soc.* 2012;3994–3997. [PubMed: 23366803]
2. Hill MD, Marty MR. Amdahl's Law in the Multicore Era. *IEEE Computer.* 2008; 41(7):33–38.
3. Manjikian N, Abdelrahman T. Scheduling of Wavefront Parallelism on Scalable Shared-memory Multiprocessors. 25th Int'l Conf on Parallel Processing. 1995
4. Levin B, Lieberman DA, McFarland B, Smith RA, Brooks D, Andrews KS, Dash C, Giardiello FM, Glick S, Levin TR, Pickhardt P, Rex DK, Thorson A, Winawer SJ. Screening and surveillance for the early detection of colorectal cancer and adenomatous polyps, 2008: a joint guideline from the American Cancer Society, the US Multi-Society Task Force on Colorectal Cancer and the American College of Radiology. *CA Cancer J Clin.* 2008; 58(3):130–60. [PubMed: 18322143]
5. Yoshida H, Näppi J. CAD in CT colonography without and with oral contrast agents: Progress and challenges. *Comput Med Imaging Graph.* 2007; 31(4–5):267–84. [PubMed: 17376650]
6. Cai W, Yoshida H, Zalis ME, Näppi J, Harris GJ. Informatics in radiology: Electronic cleansing for noncathartic CT colonography: a structure-analysis scheme. *RadioGraphics.* 2010; 30(3):585–602. [PubMed: 20219839]
7. Cai W, Holalkere N, Harris G, Sahani D, Yoshida H. Dynamic-threshold level set method for volumetry of porcine kidney in CT images: in-vivo and ex-vivo assessment of the accuracy of volume measurement. *Acad Radiol.* 2007; 14(7):890–896. [PubMed: 17574138]
8. Amdahl GM. Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capabilities. *Proc Am Federation of Information Processing Societies Conf.* 1967:483–485.

Figure 1.

(Left) Parallel software library layers in the HPC-3D platform. (Right) Illustration of the distributed block volume structure used as the fundamental data structure for distributed parallelization in the HPC-3D Platform.

Figure 2.

(a) Reduction of execution time as a function of CPU cores. (2) Speedup in HPC-EC processes as a function of the number of CPU cores.

Figure 3.

(a) Performance curve (solid line) obtained by fitting the Hill-Martin model to the speedup data of the entire HPC-EC process (red circles). (b) Performance scalability curve obtained by extrapolating the performance curve fitted in (a) up to 128 cores.

Table 1

Average execution times of the four modules in HPC-EC and sequential EC.

# of cores	EC modules				
	Colon segmentation (sec)	Structure analysis (sec)	Roughness analysis (sec)	Dynamic Level Set (sec)	Total time (sec)
Sequential EC	300	240	420	720	1680
HPC-EC	2 cores	21	67	108	181
	4 cores	14	36	47	115
	8 cores	9	21	35	90
	12 cores	8	19	32	81