Aspects of GPU for General Purpose High Performance Computing

Reiji Suda

Graduate School of
Information Science and Technology,
The University of Tokyo
& JST, CREST
e-mail: reiji@is.s.u-tokyo.ac.jp

Takayuki Aoki

Global Scientific Information
and Computing Center,
Tokyo Institute of Technology
& JST, CREST
e-mail: taoki@gsic.titech.ac.jp

Shoichi Hirasawa

Graduate School
of Information Systems,
The University of Electro-Communications
& JST, CREST
e-mail: hirasawa@is.uec.ac.jp

Akira Nukada

Global Scientific Information
and Computing Center,
Tokyo Institute of Technology
& JST, CREST

e-mail: nukada@matsulab.is.titech.ac.jp

Hiroki Honda

Graduate School of
Information Systems,
The University of Eletro-Communications
& JST, CREST

e-mail: honda@is.uec.ac.jp

Satoshi Matsuoka

Global Scientific Information
and Computing Center,
Tokyo Institute of Technology
& JST, CREST
& National Institute of Informatics
e-mail: matsu@is.titech.ac.jp

Abstract— We discuss hardware and software aspects of GPGPU, specifically NVIDIA cards and CUDA, from viewpoint of parallel computing. The major weak points of GPU against newest supercomputers are identified to be *only* four: large SIMD width, small memory, absence of fast L2 cache, and high register spill penalty. As software concerns, we derive optimal scheduling algorithm for latency hiding of host-device data transfer using divisible load theory, and discuss possible exploitation of non-SIMD parallelism on GPUs.

I. Introduction

Researchers of various fields are now paying much attention to General Purpose GPU (Graphic Processing Unit) computing, or GPGPU. Some GPU venders provide programming languages for GPGPU, such as CUDA by NVIDIA and Brook+ by AMD, and Intel Larrabee may be accompanied by some programming language for it. Although Brook+ is based on Brook, which was introduced a decade ago, CUDA seems to have somewhat better technological maturity. So our discussions in this paper will be focused on NVIDIA's GPU with CUDA programming language.

In this paper, we investigate hardware and software aspects of current GPGPU systems, and evaluate them as general purpose high performance computers.

II. Our Achievements

Before entering general discussion, some of our research achievements are introduced. These examples illustrate possibilities of GPUs in high performance computing. The terms of GPU hardware/software will be explained in the next section.

Aoki and Ogawa are using GPUs for research on computational fluid dynamics. They used 4 cards of GeForce 8800 Ultra and attained sustained performance of 51.9 Gflops for Riken benchmark size S (Poisson equation solution with point-Jacobi method), and 93.6 Gflops for size L [1]. They used the shared memory on the MP (Multi-Processor) as a user-controlled data cache to reduce performance degradation from long latency of device memory access. Their result won the first prize in single PC section of Riken benchmark contest 2007. They simulate various kinds of fluid dynamics using GPU, such as shown in Figs. 1, 2 and 3.

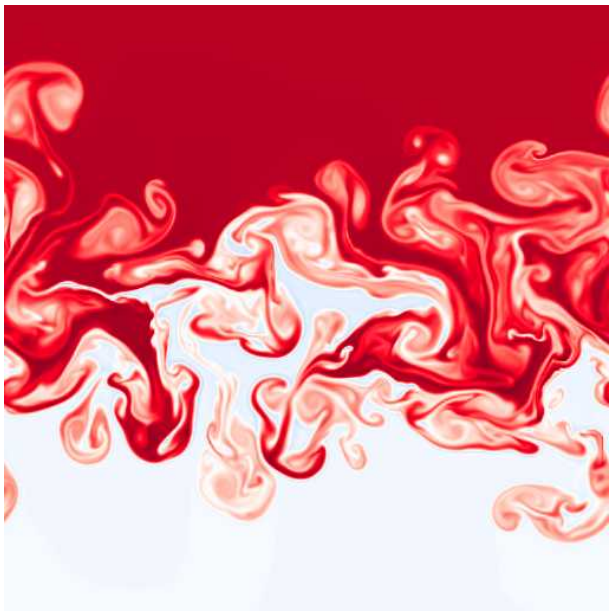

Fig. 1. GPU simulation of Rayleigh-Taylor instability of compressible fluids

Nukada, Ogata, Endo and Matsuoka [2] implemented a 3D FFT with CUDA. It achieves more than 80 Gflops for size 256^3 with GeForce 8800 GTX, which is more than three times faster than that provided by NVIDIA, CUFFT. In the presentation they also report that the performance over 300 Gflops was at-