Distributed Pararell Processing Laboratory

Stanislav G. Sedukhin
Professor

Hitoshi Oi
Assistant Professor

Naohito Nakasato
Assistant Professor

Marchin Paprzyoki
Visiting Researcher

Veles Oleksandr
Visiting Researcher

Summary of Achievement

Refereed Journal Papers

Refereed Proceeding Papers

[hitoshi-01:2012] Hitoshi Oi and Sho Niboshi. Performance and Power Consumption Measurement of Java Application Servers. In *Proceedings of IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'12)*, page 10.1109/MASCOTS.2012.68, 2012.

In this paper, we present our in-progress project of modeling performance and power consumption of Java application servers using SPECjEnterprise2010. We run the workload on two application server using two different CPUs, AMD Phenom II and Intel Atom, and investigate performance and power consumption behaviors against the increasing system sizes. We have observed that: (1) CPU utilization draws non-linear functions of the system size and their shapes are different on Phenom and Atom. However, power consumption on both servers increase proportionally. (2) Browse transaction is the source of non-linearly in the CPU utilization. (3) Estimation of the CPU utilization from that of each transaction measured separately incurs large errors (up to 65%), while the errors in the estimation of the power consumption are relatively small (up to 4%).

[hitoshi-02:2012] Hitoshi Oi and Sho Niboshi. Workload Analysis of SPECjEnterprise2010. In *Proceedings of the 10th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-12)*, page 10.1109/ISPA.2012.52, 2012.

In this paper, we present a case study of measuring the performance of SPECjEnterprise2010 and its workload analysis on two different configurations, where either the application or database server is the performance bottleneck. The CPU utilization of the application and database servers behave differently for the increasing system size. They draw square-root and quadruple like functions, respectively. By measuring each transaction type separately, we find that the source of these non-linear factors is the Browse transaction. Also, the sum of the CPU utilization of each transaction type executed individually overestimates the total CPU utilization when all transaction types are executed simultaneously. In the performance modeling methodology of SPECjAppServer2004 found in the literature, the CPU time for each transaction is assumed to be constant and it is obtained by the measurement of individual execution. However, from our observations, this methodology cannot be directly applied to SPECjEnterprise2010.

[nakasato-02:2012] N.Nakasato, H.Daisaka, T.Fukushige, A.Kawai, J.Makino, F.Yuasa, and T.Ishikawa. GRAPE-MPs: Implementation of an SIMD for quadruple/hexuple/octuple-precision arithmetic operation on a structured ASIC and an FPGA. In *2012 IEEE 6th International Symposium on Embedded Multicore/Many-core System-on-Chip*, pages 75–83, 2012.

We describe the design and performance of the GRAPE-MPs, a series of SIMD accelerator boards for quadruple/hexuple/octuple-precision arithmetic operations. Basic design of GRAPE-MPs is that it consists of a number of processing elements (PE) and memory components which handle data with quadruple/hexuple/octuple-precision. A GRAPE-MPs processor is implemented on a structured ASIC chip and an FPGA chip. GRAPE-MP (quadruple-precision) uses a structured ASIC chip from eASIC corp., which has 6 PE and operates with 100MHz clock cycle. The theoretical peak quadruple-precision performance of the single board is 1.2 Gflops and the achieved performance for the Feynman loop integrals is about 0.5 Gflops. GRAPE-MP4/6/8 (quadruple/hexuple/octuple-precision) uses an FPGA chip from Aletra corporation. For example, in the current implementation, MP8 has 10 PE with 70MHz operation clock cycle. We also present the performance results with the multiple GRAPE-MPs boards. The achieved performance of four MP8 boards is about 1.6 Gflops. It is roughly 90 times faster than the performance of a single core of a CPU with comparable precision. We show that our hardware based approach to evaluate the Feynman loop integrals in high precision arithmetic operations is highly effective.

[nakasato-03:2012] K.Matsumoto, N.Nakasato, and S.G.Sedukhin. Implementing a Code Generator for Fast Matrix Multiplication in OpenCL on the GPU. In *2012 IEEE 6th International Symposium on Embedded Multicore/Many-core System-on-Chip*, pages 198–204, 2012.

This paper presents results of an implementation of code generator for fast general matrix multiply (GEMM) kernels. When a set of parameters is given, the code generatorproduces the corresponding GEMM kernel written in OpenCL. The produced kernels are optimized for high-performanceimplementation on GPUs from AMD. Access latencies to GPU global memory is the main drawback for high performance. This study shows that storing matrix data in a block-major layout increases the performance and stability of GEMM kernels. On the Tahiti GPU (Radeon HD 7970), our DGEMM (double-precision GEMM) and SGEMM (single-precision GEMM) kernels achieve the performance up to 848 GFlop/s (90% of the peak) and 2646 GFlop/s (70%), respectively.

[nakasato-04:2012] K.Matsumoto, N.Nakasato, and S.G.Sedukhin. Performance Tuning of Matrix Multiplication in OpenCL on Different GPUs and CPUs. In *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion: 3rd International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems*, pages 396–405, 2012.

OpenCL (Open Computing Language) is a framework for general-purpose parallel programming. Programs written in OpenCL are functionally portable across multiple processors including CPUs, GPUs, and also FPGAs. Using an auto-tuning technique makes performance of OpenCL programs also portable on different processors. We have developed an auto-tuning system with a code generator for fast matrix multiply kernels in OpenCL. This paper presents results of performance evaluation of DGEMM (double-precision general matrix multiply) and SGEMM (single-precision GEMM) implementations by using the auto-tuning system. Performance evaluations are conducted on two AMD GPUs (Tahiti and Cayman), two NVIDIA GPUs (Kepler and Fermi), and two CPUs (Intel Sandy Bridge and AMD Bulldozer). Our GEMM implementations on the AMD GPUs show higher performance than the highly tuned vendor library while the implementations on the NVIDIA GPUs are comparable.

[sedukhin-02:2012] K. Matsumoto, N. Nakasato, and S.G. Sedukhin. Performance Tuning of Matrix Multiplication in OpenCL on Different GPUs and CPUs. In *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, pages 396–405, Salt Lake City, November 2012. ACM, IEEE Computer Society Press.

OpenCL (Open Computing Language) is a framework for general-purpose parallel programming. Programs written in OpenCL are functionally portable across multiple processors including CPUs, GPUs, and also FPGAs. Using an auto-tuning technique makes performance of OpenCL programs also portable on different processors. We have developed an auto-tuning system with a code generator for fast matrix multiply kernels in OpenCL. This paper presents results of performance evaluation of DGEMM (double-precision general matrix multiply) and SGEMM (single-precision GEMM) implementations by using the auto-tuning system. Performance evaluations are conducted on two AMD GPUs (Tahiti and Cayman), two NVIDIA GPUs (Kepler and Fermi), and two CPUs (Intel Sandy Bridge and AMD Bulldozer). Our GEMM implementations on the AMD GPUs show higher performance than the highly tuned vendor library while the implementations on the NVIDIA GPUs are comparable.

[sedukhin-03:2012] K. Matsumoto, N. Nakasato, and S.G Sedukhin. Performance Tuning of Matrix Multiplication in OpenCL on Different GPUs and CPUs. In *Embedded Multicore SoCs (MCSoC), 2012 IEEE 6th International Symposium on*, pages 198–204, Aizuwakamatsu, September 2012. ACM, IEEE Computer Society Press.

This paper presents results of an implementation of code generator for fast general matrix multiply (GEMM) kernels. When a set of parameters is given, the code generator produces the corresponding GEMM kernel written in OpenCL. The produced kernels are optimized for high-performance implementation on GPUs from AMD. Access latencies to GPU global memory is the main drawback for high performance. This study shows that storing matrix data in a block-major layout increases the performance and stability of GEMM kernels. On the Tahiti GPU (Radeon HD 7970), our DGEMM (double-precision GEMM) and SGEMM (single-precisionGEMM) kernels achieve the performance up to 848 GFlop/s (90% of the peak) and 2646 GFlop/s (70%), respectively.

[sedukhin-04:2012] Stanislav Sedukhin and Marcin Paprzycki. Generalizing Matrix Multiplication for Efficient Computations on Modern Computers. In Roman Wyrzykowski, Jack Dongarra, Konrad Karczewski, and Jerzy Waśniewski, editors, *Parallel Processing and Applied Mathematics*, Lecture Notes in Computer Science, pages 225–234. Springer Berlin Heidelberg, 2012.

Recent advances in computing allow taking new look at matrix multiplication, where the key ideas are: decreasing interest in recursion, development of processors with thousands (potentially millions) of processing units, and influences from the Algebraic Path Problems. In this context, we propose a generalized matrix-matrix multiply-add (MMA) operation and illustrate its usability. Furthermore, we elaborate the interrelation between this generalization and the BLAS standard.

[sedukhin-05:2012] Tomoya Sakai, Naohito Nakasato, and Stanislav Sedukhin. 3-Dimensional Linear Transforms with Cubical Data Decomposition in

Torus Network. In Les Miller, editor, *Proc. of the The 28th International Conference on Computers and Their Applications (CATA-2013)*, pages 99–106, Honolulu, Hawaii, USA, March 2013. International Society of Computers and Their Applications (ISCA).

This paper presents implementation and performance evaluation of the three-dimensional (3D) $N \times N \times N$ forward/inverse discrete transforms in torus network of computer nodes. We implemented newly proposed GEMM-based algorithm with 3D data decomposition which can be extremely scaled up to N^3 computer nodes. We propose the way of overlapping of computing and communication which leads to the performance improvement. The performance evaluation of the algorithm with 3D data decomposition on the RIKEN Integrated Cluster of Clusters (RICC) is presented.

Academic Activities

[hitoshi-03:2012] Hitoshi Oi, Since 2005.

Professional Member, ACM

[hitoshi-04:2012] Hitoshi Oi, Since 2005.

Member, IEEE/Computer Society

[hitoshi-05:2012] Hitoshi Oi, Since 2006.

Academic Member, EEMBC http://eembc.org/

[hitoshi-06:2012] Hitoshi Oi, Since 2009.

Senior Member, IACSIT http://www.iacsit.org/

[hitoshi-07:2012] Hitoshi Oi, 2012.

Program committee member and chair of the special session in Network on Chip and Multi-core technologies (NMT2012).

[hitoshi-08:2012] Hitoshi Oi, 2012.

Program Committee Member and Session Chair (Track 2 Architectures and virtualization).

[hitoshi-09:2012] Hitoshi Oi, 2012.

Session chair (Executin Tools)

[hitoshi-10:2012] Hitoshi Oi, Since 2010.

Officer

[hitoshi-11:2012] Hitoshi Oi, Since 2009.

Academic member of the T-Engine Forum (representative for the University of Aizu). http://www.t-engine.org/

Ph.D and Others Theses

[nakasato-05:2012] Ryo Tayama. Graduation Thesis: Data Compression of Huffman Encoding on GPU, University of Aizu, 2013.

Thesis Advisor: N.Nakasato

[sedukhin-06:2012] Kazuya Matsumoto. Doctoral Dissertation: Design and performance optimization of matrix multiplication and shortest-path algorithms on hybrid CPU/GPU systems, University of Aizu, 2013.

Thesis Advisor: S. Sedukhin

[sedukhin-07:2012] Tomoya Sakai. Master Thesis: 3D Discrete Transform with Cubical Data Decomposition in Torus Network, University of Aizu, 2013.

Thesis Advisor: S. Sedukhin

Others

[hitoshi-12:2012] Hitoshi Oi.

Journal reviewer for Microprocessor and Microsystems (Elsevier) and International Journal of High Performance Systems Architecture (Inderscience Enterprises)