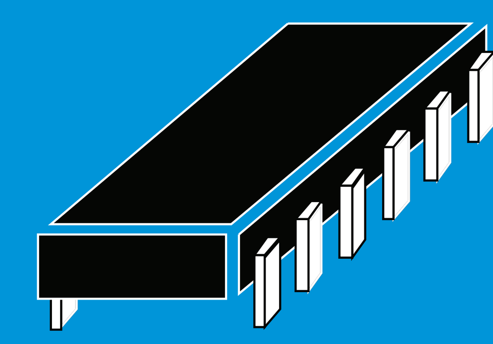
Center for Computational Sciences, University of Tsukuba
www.ccs.tsukuba.ac.jp



High Performance Computing Research

High Precision BLAS on GPUs

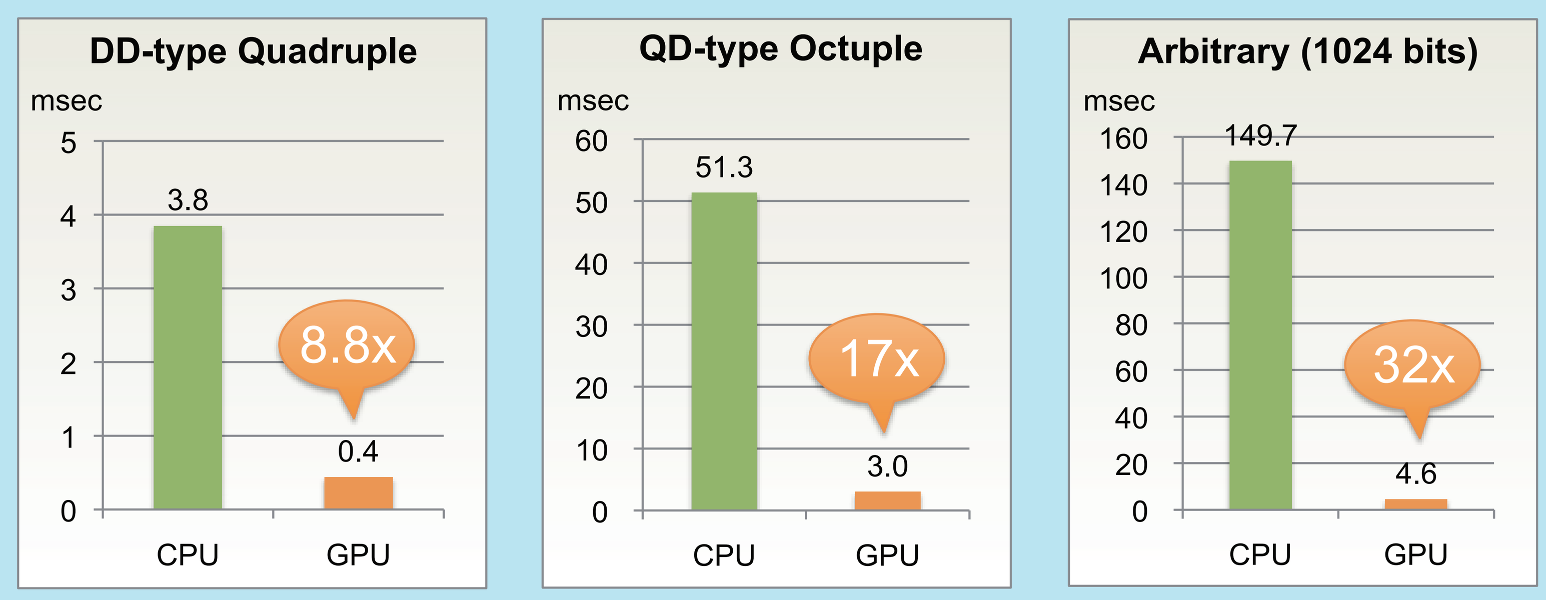
- There is high demand for high precision computations. e.g. reducing error accumulation in large scale computing
- Since high precision computations are computationally complex, most traditional processors cannot perform them efficiently.
- It's possible to quickly execute high precision BLAS operations on GPUs.
- We implemented some BLAS functions on GPUs and compared their performance to the same functions on CPUs.

Overview

- Implemented using CUDA for NVIDIA's GPUs
- Supports three type of high precision computations
  - 1. Double-double (DD) type quadruple precision
    - 106 bits significand (approx. 32 decimal digits)
    - Stores quadruple precision values into two double precision floating point values
  - 2. Quad-double (QD) type octuple precision
    - 212 bits significand (approx. 64 decimal digits)
    - Stores octuple precision values into four double precision floating point values
  - 3. Arbitrary precision (variable length)
    - GNU MP compatible
    - Stores high precision values into a format defined by integer values

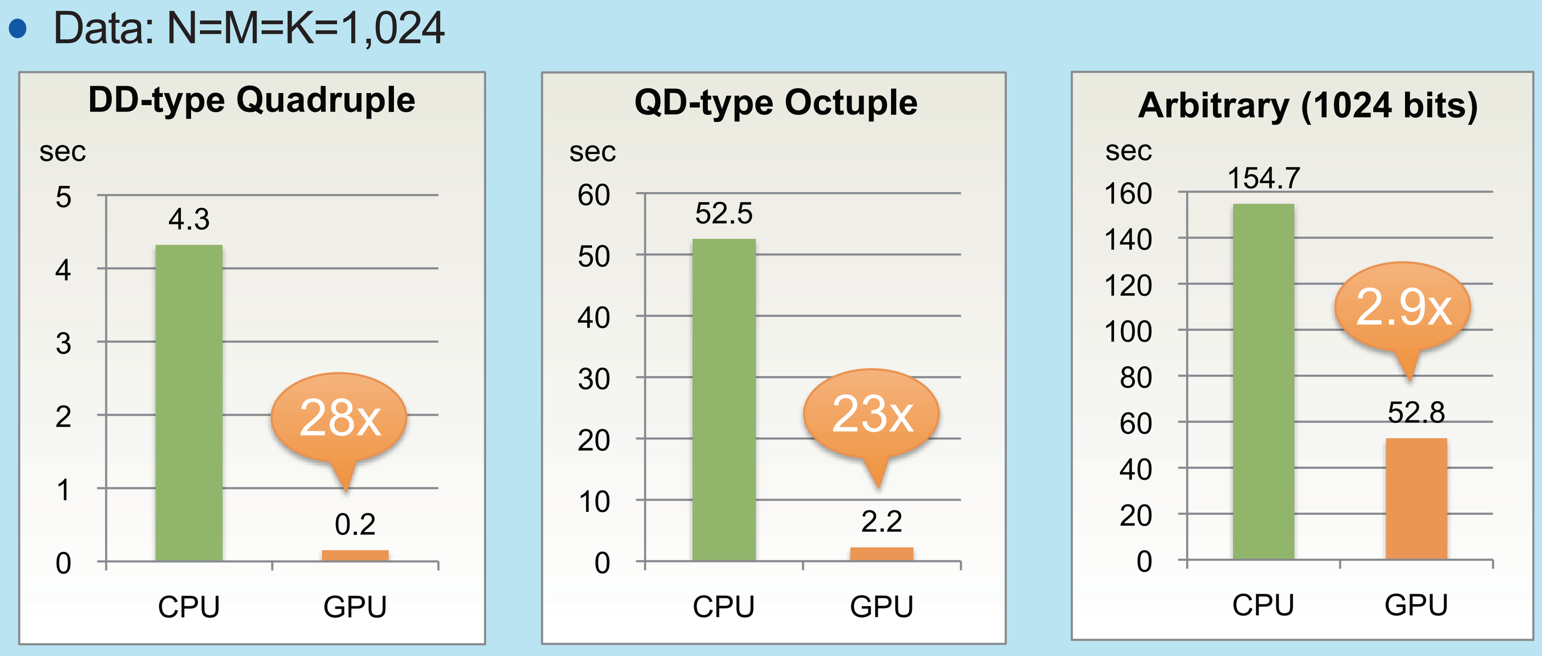
Performance: AXPY

- GPU: NVIDIA Tesla C2050 (Fermi architecture, ECC-disabled)
- CPU: Intel Core i7 920 (2.67 GHz, Quad-Core, HT-disabled)
- Data: N=1,024,000



Performance: GEMM

- GPU: NVIDIA Tesla C2050 (Fermi architecture, ECC-disabled)
- CPU: Intel Core i7 920 (2.67 GHz, Quad-Core, HT-disabled)
- Data: N=M=K=1,024



FFTE: A High-Performance FFT Library

- FFTE is a Fortran subroutine library for computing the Fast Fourier Transform (FFT) in one or more dimensions.
- It includes complex, mixed-radix and parallel transforms.
- FFTE is typically faster than other publicly-available FFT implementations, and is even competitive with vendor-tuned libraries.

Features

HPC Challenge benchmark

- High speed
  - Supports Intel's SSE2/SSE3 instructions.
- Parallel transforms
  - Shared / Distributed memory parallel computers (OpenMP, MPI and OpenMP + MPI)
- High portability
  - Fortran77 + OpenMP + MPI
  - Intel's intrinsics for SSE2/SSE3 instructions.
- HPC Challenge Benchmark
  - FFTE's 1-D parallel FFT routine has been incorporated into the HPC Challenge (HPCC) benchmark.

Design

- Performance
  - One goal for large FFTs is to minimize the number of cache misses.
- Ease of use: routine interfaces
  - Similar to sequential SGI SCSL or Intel MKL routines
- Portability
  - Communication: MPI
  - Computation: Fortran77 + OpenMP

Approach

- Many FFT routines work well when data sets fit into cache.
- When the problem size exceeds the cache size however, the performance of these FFT routines decreases dramatically.
- Some previously presented six-step FFT algorithms require
  - Two multicolumn FFTs.
  - Three data transpositions: these are the chief bottlenecks in cache-based processors.
- We have combined the multicolumn FFTs and transpositions to reduce the number of cache misses.

Performance of FFTE 4.0

- Data:
  - N1 x N2 x N3 = 2^24 x P
- Machines:
  - Xeon EM64T 3.0 GHz
  - Gigabit Ethernet
  - 1024 MB DDR2/400