



# NukadaFFT

## An auto-tuning FFT library for CUDA GPUs

Akira Nukada<sup>1,3</sup> and Satoshi Matsuoka<sup>1,2,3</sup>

1. Tokyo Institute of Technology 2. National Institute of Informatics 3. Japan Science and Technology Agency

### Introduction

We have released our FFT library for CUDA GPUs at the following URL.

<http://matsu-www.is.titech.ac.jp/~nukada/nufft/>

Most of algorithms and auto-tuning technologies of FFT for CUDA are already presented [1,2], and the performance is much higher than NVIDIA CUFFT library. Several parameters of GPU FFT such as factorization, number of threads, and padding insertion pattern are automatically tuned. The library now supports new Fermi architecture and works with CUDA 3.0 or later.

[1] "Bandwidth Intensive 3-D FFT algorithm for CUDA GPUs", SC'08, Austin, 2008.

[2] "Auto-Tuning 3-D FFT library for CUDA GPUs", SC'09, Portland, 2009.

Initial version of our library is designed for G8X/G9X and GT200 series.

To support Fermi cores, we need only small modifications as follows.

(1) Changed CUDA PTX version from 1.4 to 2.0

This modification is limited in declaration lines.

(2) Added cache control options for each global load/store instructions.

Global memory access is cacheable in Fermi. We have to choose the best cache options. The option is now statically selected.

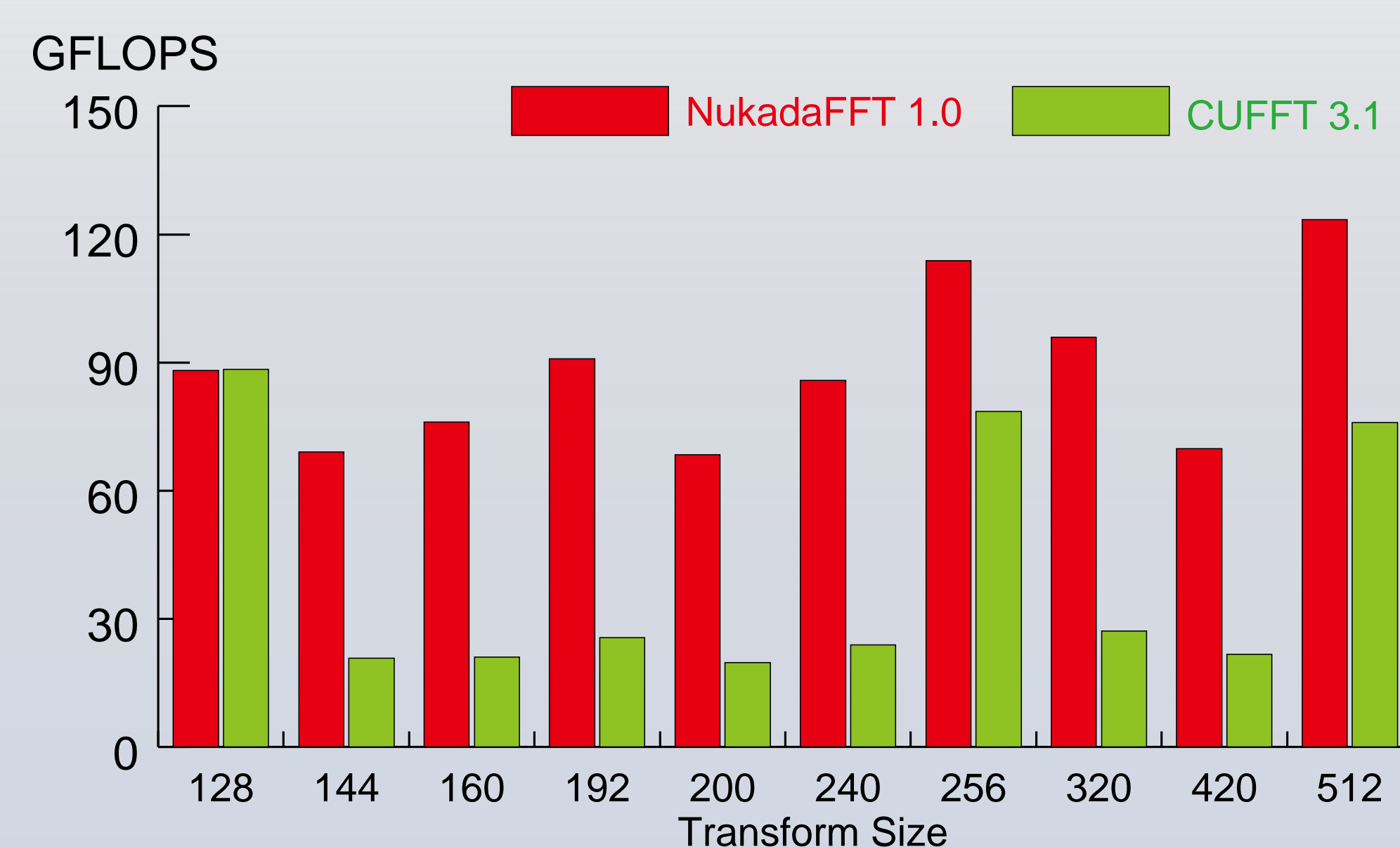
(3) Replaced `shared memory overrun` with `predicated execution`.

In the CUDA implementations of FFT, threads may access the shared memory conditionally. To avoid the branch instructions, our initial version used the shared memory overrun technique, which was faster than the predicated execution in older GPUs. However, Fermi core does not allow the technique.

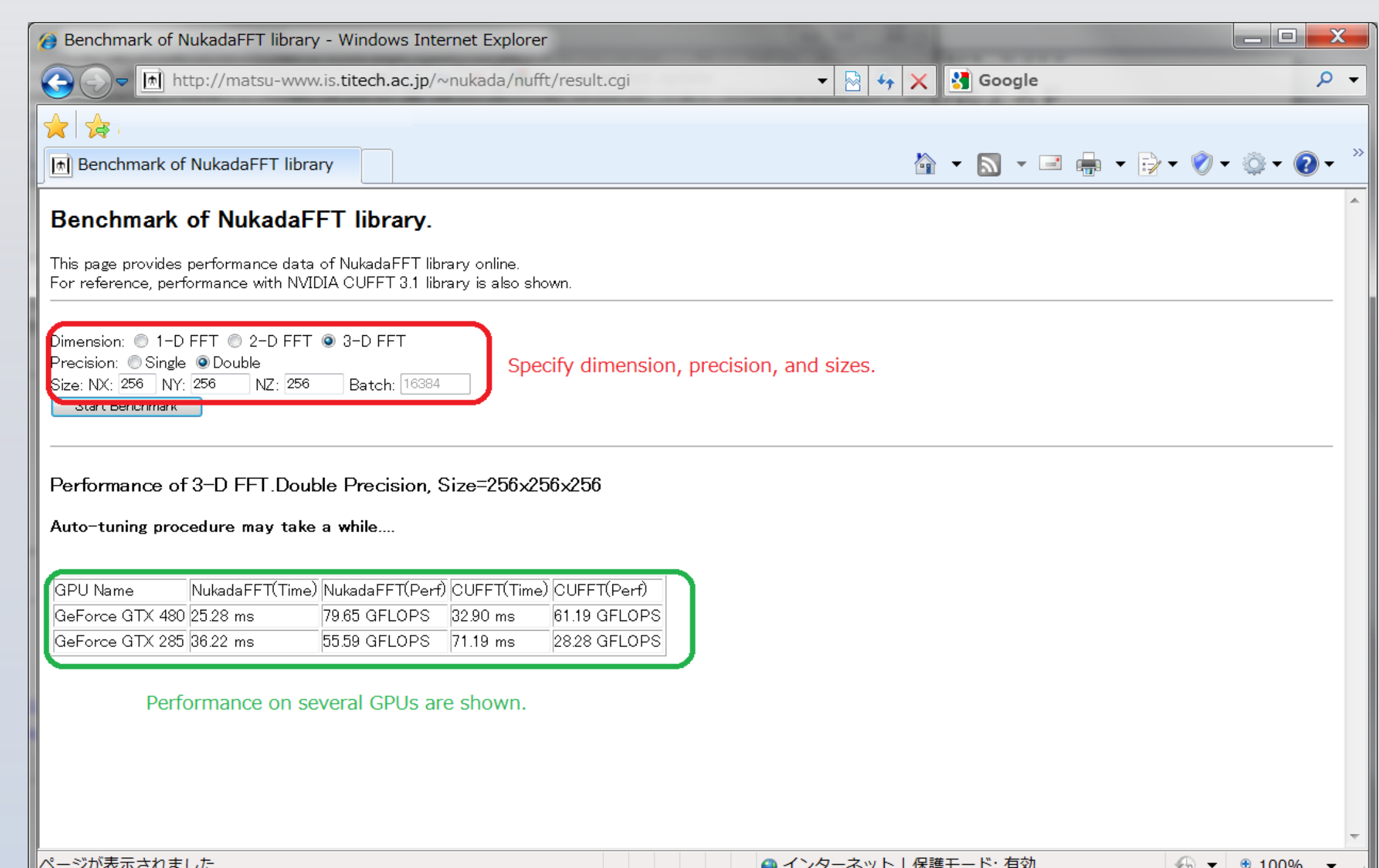
### Fermi Support

Our website provides an online benchmarking service. You can see the performance of our library for the size, dimension, and precision you are interested in, before downloading software, before buying GPUs.

### Performance



Performance of 1-D FFT.  
(Double Precision, batch=32,768, GPU=GeForce GTX 480.)



### Acknowledgment

This work was supported by the following projects:

- JST-CREST: "ULP-HPC: Ultra Low-Power, High-Performance Computing via Modeling and Optimization of Next Generation HPC Technologies".
- Microsoft Technical Computing Initiative: "HPC-GPGPU: Large-Scale Commodity Accelerated Clusters and its Application to Advanced Structural Proteomics".
- NVIDIA CUDA COE (Tokyo Tech).
- MEXT Grant-in-Aid for Young Scientists (A) 22680002.