Quantifying Power Efficiency of FFTs on NVidia GPUs

Document Number ..... Not Defined
Document Type ..... Memo
Revision 1B
Author ..... J. Kent, B. Nikolic
Release Date 2016-11-14
Document Classification ..... Unrestricted
Status ..... Draft

Lead Author	Designation	Affiliation
James Kent	Research Student	University of Cambridge
Signature & Date:		

Owned by	Designation	Affiliation
Bojan Nikolic	Project Engineer	University of Cambridge
Signature & Date:		

Approved by	Designation	Affiliation
Name of the approver - to be filled in	Designation - to be filled in	Affiliation - to be filled in
Signature & Date:		

Released by	Designation	Affiliation
Paul Alexander	SDP Project Lead	University of Cambridge
Signature & Date:		

Revision	**Date of issue**	**Prepared by**	**Comments**
1B			

ORGANISATION DETAILS

Name	Science Data Processor Consortium
Address	Astrophysics Cavendish Laboratory JJ Thomson Avenue Cambridge CB3 0HE
Website	http://ska-sdp.org
Email	ska-sdp-pa@mrao.cam.ac.uk

Table of Contents

List of Figures

List of abbreviations

DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FLOPS	Floating Point Operations Per Second
GPU	Graphical Processing Unit

1 Introduction

Capabilities of the Science Data Processor will be limited by both the capital and operational (especially energy cost) budgets. In the current SDP costing Graser et al. (2016) both the quantity of hardware needed for the main computational steps and its power usage are inversely proportional to a single number which is the 'computational efficiency'. These two estimates then feed into both the capital and operational budgets in a major way.

In this document we expand on a perhaps under-appreciated point that energy efficiency (i.e., the ratio of theoretical minimum to actual energy to compute the result) tends to be higher than computational efficiency (i.e., the ratio theoretical minimum to actual processor$\times$hours to compute the result) on modern processing hardware which can power-down or down-clock unused parts of the processor. We measure and quantify this effect on NVidia GPUs which are used as the basis of the costing in Graser et al. (2016). The measurements done on one representative kernel, the 2D FFT, only.

2 Measurement

2.1 Objectives

The primary objective was to measure the power used during the computation of FTTs on NVidia GPUs and use this to estimate actual energy efficiency of the GPUs and contrast this with worst-case energy efficiency. To make this estimate we measured the on-card power draw and kernel execution time for the FFTs for a range of sizes and dimensions.

2.2 Experimental Hardware

We made the measurements on an Nvidia Tesla K20c card with 5.1 GB of GDDR5 memory. This card has a theoretical peak performance of 3.5 TFLOPS for single precision floating point and 1.17 TFLOPS for double precision. The nominal TDP of the card is 225W.

The card includes circuitry to measure the actual power draw which we used in the experiment. This is not a calibrated meter so the power measurements are be assumed to be accurate to +/- 5 W. The meter does not estimate the system power draw, which is addressed in the discussion section below.

The GPU was hosted in a node with two Intel Xeon E5-2640 CPUs and 32GB DDR3 memory. The node was dedicated for this testing for the duration of the measurements.

2.3 Software

The operating system was CentOS Linux Version 7.2.1511 with NVidia Driver 352.39. The FFTs were computed using CUDA version 7.5.18. A Python script was used orchestrate the experiment and collect the data.

2.4 Experiment Runs

Measurements were done for a variety of input sizes for Complex-to-Complex transforms, including sizes which are products of prime sizes 2^N , 3^N, 5^N, 7^N, 11^N as well as non-prime factors. All of 1-dimensional, 2-dimensional, and 3-dimensional transforms were measured but concentrating on the 2-dimensional transform most relevant for SDP. Each size of FFT was executed 10 times and averaged to account for any variation in the execution time on the GPU. The quantities recorded were the execution time and high-timer resolution and the power draw (for some of the runs).

The maximum size of the transforms was limited by the available memory on the GPU. We were not able in time to obtain dedicated access for testing on GPUs with larger on-board memories.

To understand how the power draw of the card scales as a function of FFT size and factoring, each FFT size was ran continuously in a loop, and the power reading was measured using the onboard wattmeter, through the NVIDIA driver. The power efficiency was then calculated by taking the floating point performance values calculated previously, and dividing them by the power consumption to get a an efficiency result in terms of performance per unit power.

NVidia GPUs have an option in the driver to limit the maximum power draw of the card. The results presented later in this document were measured without this restriction enabled, but we also repeated many of the runs this with this restriction set to 150 W (which is the minimum the card supports). In these cases we measured performance which was indistinguishable from measurements without the power restriction.

3 Analysis

We computed the computational performance by:

$$R_{\mathrm{FLOPS}} = \frac{5N\log(N)}{t} \tag{1}$$

where N is the total number of elements input into the FFT transform and t is the execution time of the FFT kernel. This uses a theoretical estimate of the required operation count as, unlike for modern CPUs, measuring actual number of operations carried out using performance counters is not easy on GPUs.

We estimated the energy usage and energy efficiency by computing:

$$S_{\mathrm{Energy}} = t\,\langle P\rangle \tag{2}$$

$$R_{\mathrm{Energy}} = \frac{R_{\mathrm{FLOPS}}}{\langle P\rangle} \tag{3}$$

where $\langle P\rangle$ is the average power draw during the kernel execution, t is again the execution time, S_{Energy} is the energy to compute the result and R_{Energy} is the energy efficiency as number of operations per unit energy.

4 Results

The results are presented as functions of the input size of the FFT.

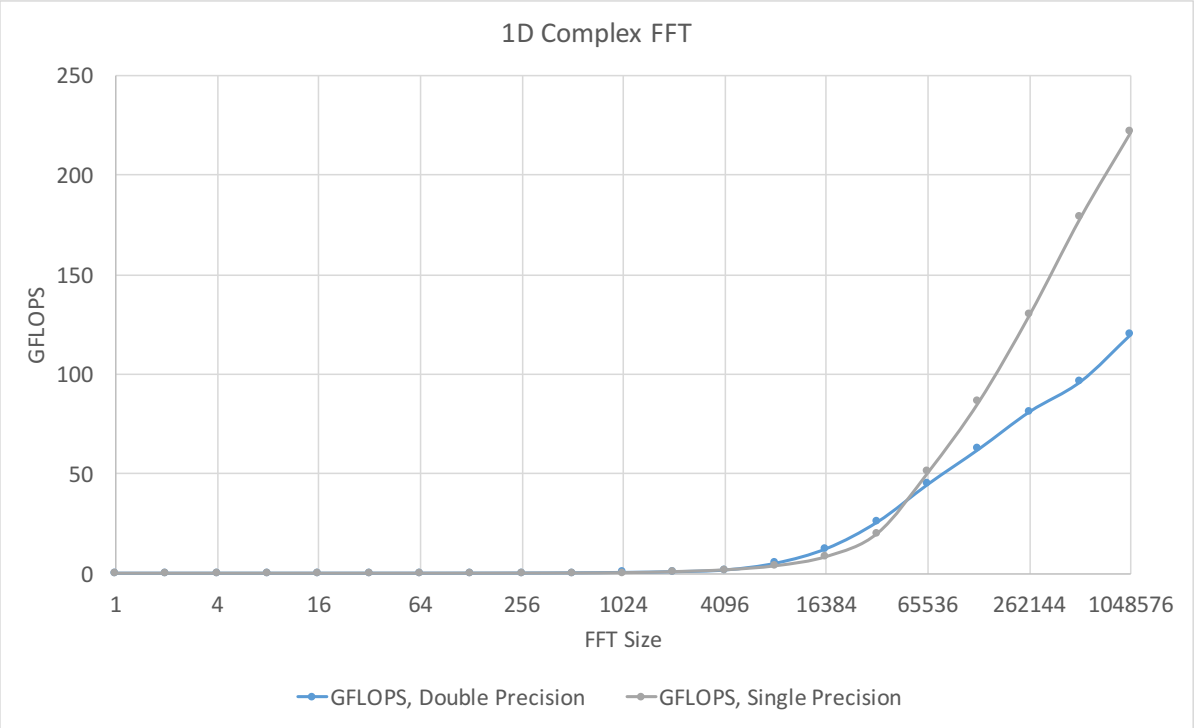

Figure 1: Performance for 1D Complex FFT of 2^N sizes

4.1 Computational Performance

Figures 1, 2 and 3 show the collected results of how FFT size influences the floating point operations per second for up to 3-Dimensional FFTs.

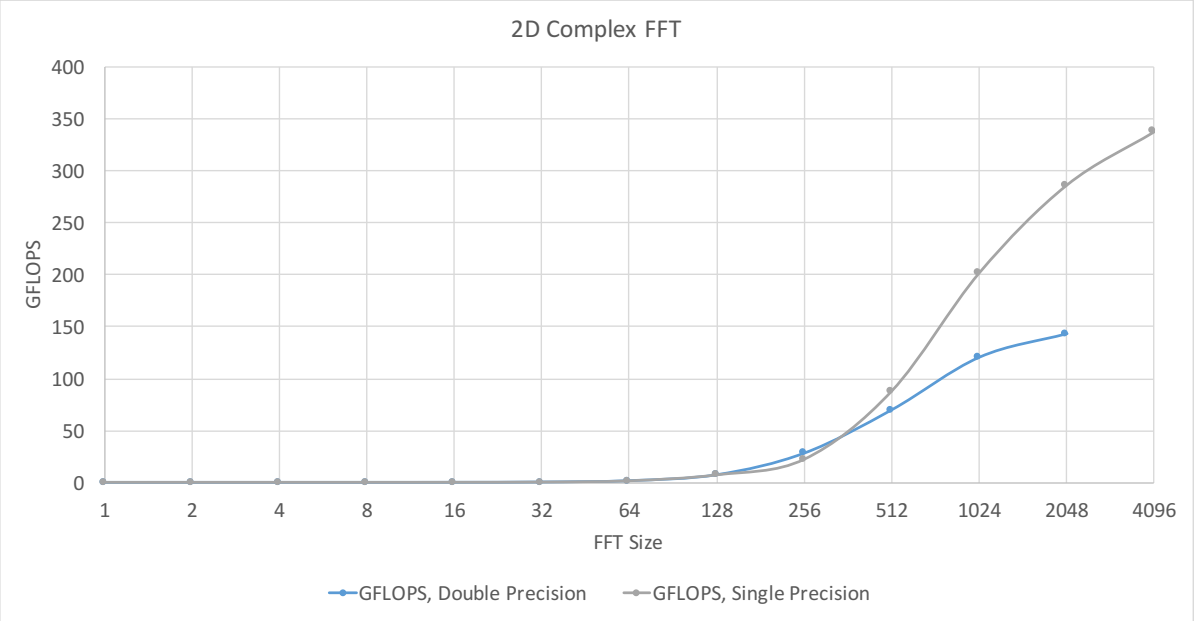

Figure 2: Performance for 2D Complex FFT of 2^N sizes

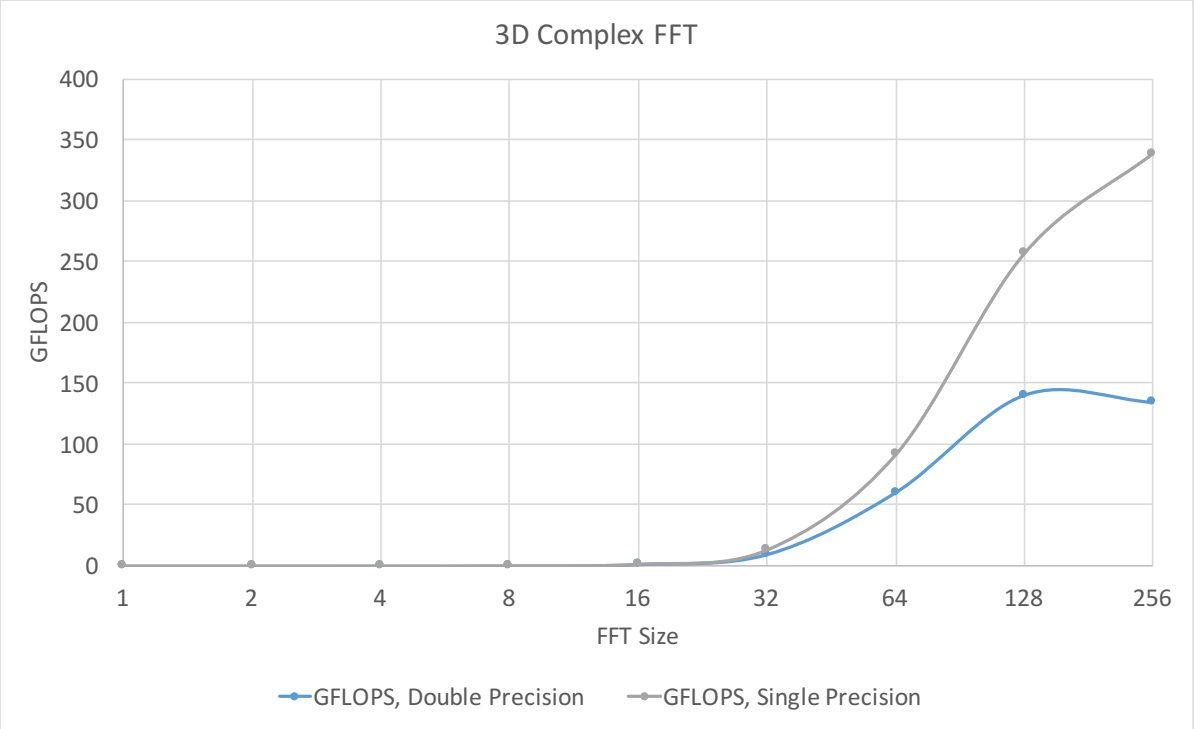

Figure 3: Performance for 3D Complex FFT of 2^N sizes

4.2 Power Efficiency

The power draw of the NVIDIA card under different workloads is shown in Figure 4, and the power efficiency shown in Figure 5.

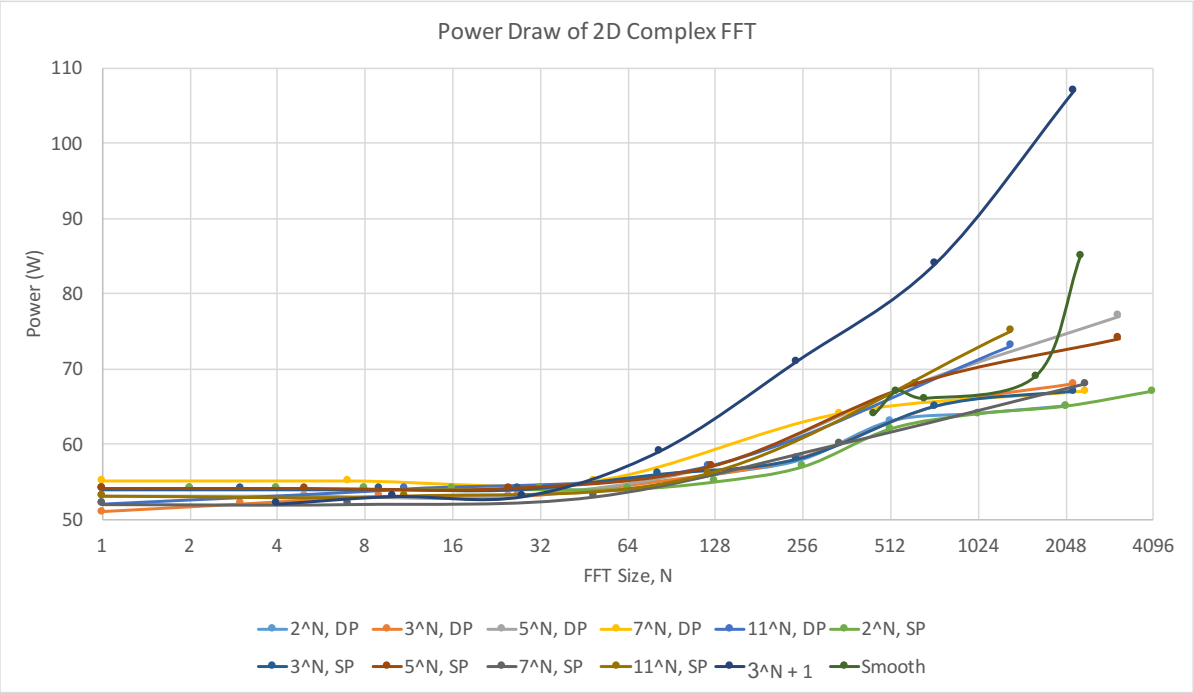

Figure 4: Power Draw for various FFT sizes

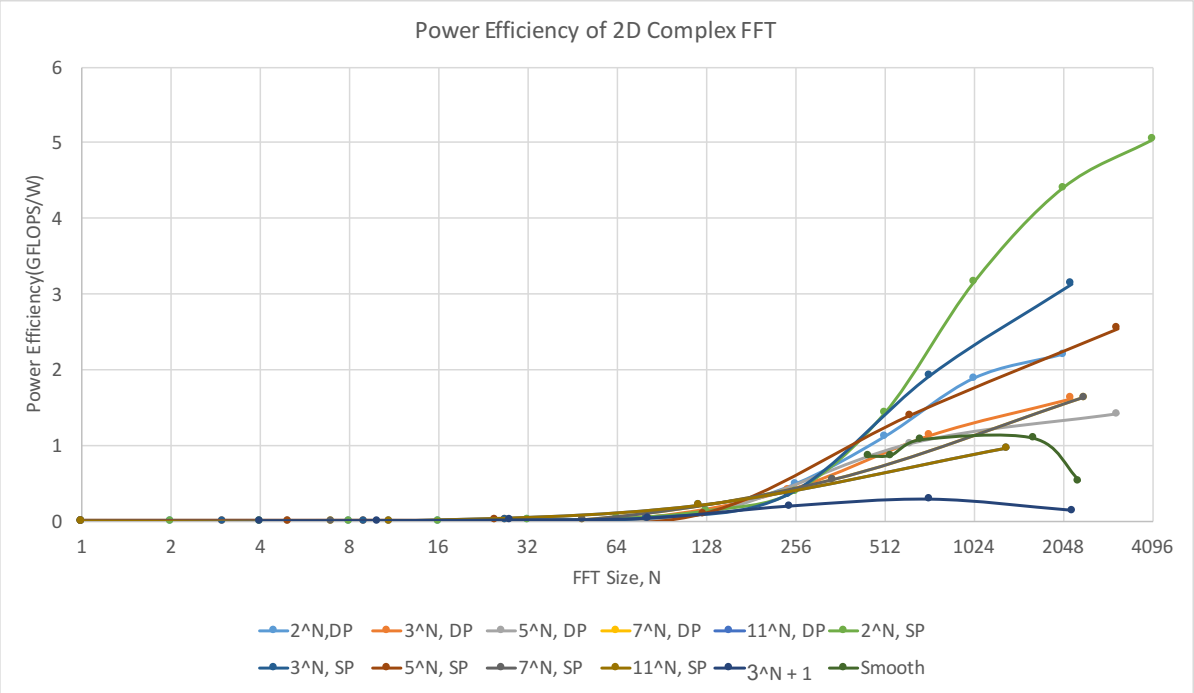

Figure 5: Power Efficiency for various FFT sizes

5 Analysis

As expected, transforms of size 2^N have the best power efficiency. N-smooth sizing such as $3^N 5^N 7^N 11^N$ have efficiencies around half of the 2^N sizes. The efficiency of double-precision transforms is half of the single-precision transforms.

Measured energy efficiencies are significantly higher than would be estimated by using the card TDP rather than the measured power draw. This is illustrated in Figure 6, where we contrast the power efficiency of a 2D double precision FFT calculated using the measured power with the efficiency using the full TDP is used.

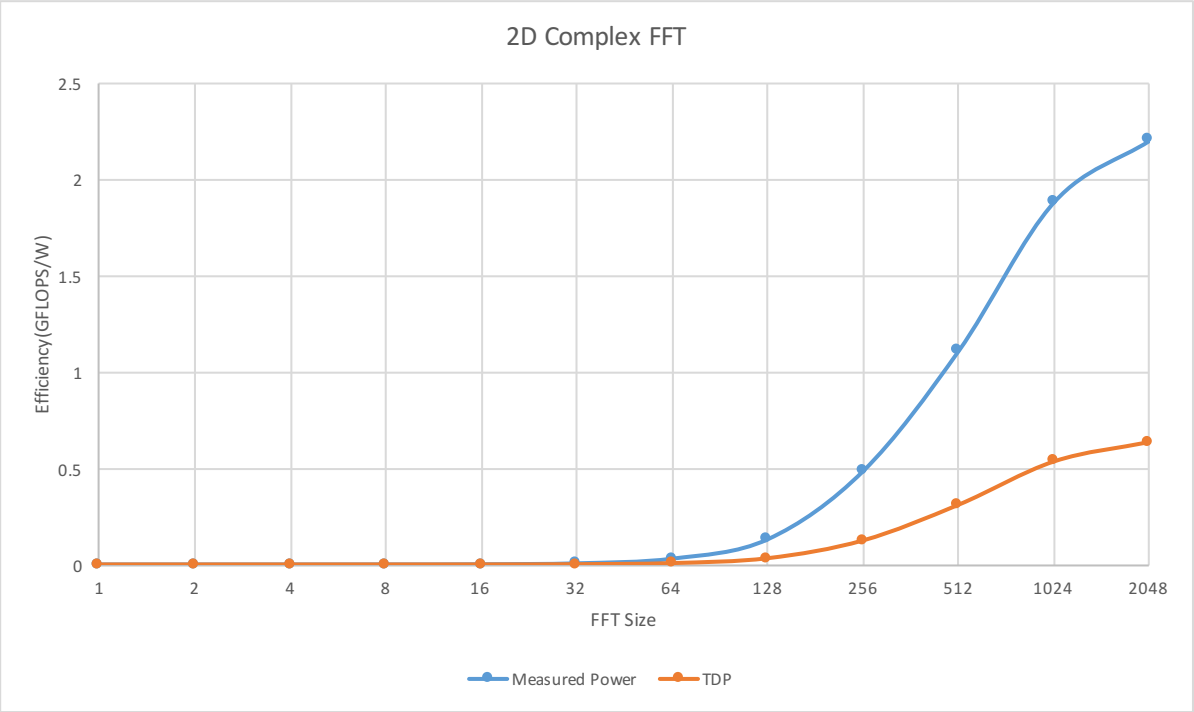

Figure 6: Power efficiency using measured power versus TDP

6 Discussion

The focus of this memo is on actual power efficiency of FFTs compared to efficiency estimated by assuming the power draw is always the TDP limit. We find that actual efficiency is up to a factor of 4 higher for 2D FFTs, a key step in SDP image reconstruction.

The current SDP power estimate is based on each GPU continuously using 300 W of power. In the light of the above this assumption should be revisited. The costing model already has an easy way to do this by changing the underclocking factor cell B5 on Compute Island hardware sheet. Experiment with restricting the power usage of the GPU to 150 W through the driver shows that peak power usage may be reduced as well as the average energy usage.

An accurate average power per GPU to be used can be derived only after all of the major computational steps are profiled in this same way. But, we believe an assumption of 50% power usage is probably conservative given the low assumed computational efficiency. In the current model, and taking into account power used in host systems, etc., this gives a 40% power saving in each compute island.

We measure the difference of power efficiency of different sizes of FFTs and find significant difference between 2^N sizes and other sizes, even products of other small primes.

Acknowledgement

We thank Kate Clark for the raising the difference of computational versus power efficiency and the suggestion to carry out the power-limiting experiment.

References

Graser, F., Taylor, J., et al.: SDP Cost Model 03-09-2016 Spreadsheet, Tech. Rep. SKA-TEL-SDP-0000043, SDP Consortium, 2016.