Journal of Computer Graphics Techniques Vol. 1, No. 1
http://jcgt.org 2012

Performance per What?

Tomas Akenine-Möller and Björn Johnsson
Intel Corporation and Lund University

Abstract

In this short note, we argue that *performance per watt*, which is often cited in the graphics hardware industry, is *not* a particularly useful unit for power efficiency in scientific and engineering discussions. We argue that joules per task and watts are more reasonable units. We show a concrete example where nanojoules per pixel is much more intuitive, easier to compute aggregate statistics from, and easier to reason about.

1. Introduction

Power and energy efficiency is one of the most important optimization axes for graphics [Johnsson et al. 2012], in particular, for mobile phones [Akenine-Möller and Ström 2003], tablets, and laptops, but also for desktop graphics [NVIDIA 2012]. Predicting or estimating power for a certain graphics algorithm running on a particular graphics processor is extremely difficult. Therefore, it makes sense to *measure* the current(s) of the graphics processor, which can be translated to power consumption, since voltage to a discrete GPU or to the motherboard of an integrated CPU/GPU is constant [Johnsson et al. 2012].

Currently, public energy-efficiency research in graphics is in its infancy, but since we believe that reporting energy and/or power consumption in future graphics research will become as common as it is to report performance (milliseconds per frame) today, it is important to settle on which units be used to report results.

2. Unit Motivation

The quantity, "performance per watt," is often used with the objective to reach as high a number as possible. Hence, we conclude that performance can be given in frames per second, for example, denoted F/t, where F is the number of frames rendered in t seconds. It is well known that $P = E/t$, where P is power measured in watts (W), E is energy measured in joules (J), and t is time measured in seconds (s). This means that performance per watt becomes

$$\frac{F/t}{P} = \frac{F \cdot t}{t \cdot E} = \frac{F}{E},$$

with unit given by *frames per joule*, actually an energy measure and not a power measure, as "performance per watt" implies. Frames per joule (fpJ) is similar to *frames per second* (fps) in that it is an inverse measure. Note that it sounds better to report in these inverse measures, e.g., getting a 3× performance improvement in fps may "sound" better than reducing the frame time by 67% . This explains the popularity for this form by marketing departments. It should also be noted that these inverse measures (fps and fpJ) cannot be averaged. So if fps or fpJ is measured over an animation, it is incorrect to average over all of the frames' fps (or fpJ) in order to compute an average fps or fpJ. Instead, the numbers should be inverted (1/fps and 1/fpJ), and then these numbers need to be averaged. Finally, one may convert back to fps or fpJ.

For similar reasons that frame time is more useful when reporting performance, frame energy (joules per frame) is more useful for reporting energy efficiency. We recently proposed the following measurement [Johnsson et al. 2012]:

$$E = \frac{\int_0^{t_{\text{tot}}} P(t)\mathrm{d}t}{F \cdot R},$$

where t_{tot} is the total time it takes to render the entire animation, F is the number of frames in the animation, and R is the screen resolution in pixels. The unit is joules per pixel (Jpp), or nanojoules per pixel (nJpp), to make the decimals more manageable.

Hypothetically, one can creative an "efficient" architecture that uses very little power, but takes a very long time to render the frame. Such an architecture would be energy efficient, but not very usable. Therefore, we argue that both frame time and energy per pixel (or energy per frame) be reported at the same time. Conversely, one can also report energy per frame divided by time per frame, which gives us a strict power measure ($P = E/t$).

3. Example

We now show an example, where we have measured power using the methodology by Johnsson et al. [2012], on an Intel Ivy Bridge processor with an HD Graphics 4000 integrated GPU. We measure on a deferred-shading renderer consisting of a first pass that creates the G-buffers (depth, position, diffuse texture, etc), and a second pass that loops over light sources in order to accumulate lighting. Similar to Johnsson et al. [2012], idle power of the CPU is subtracted by measuring the power of the program with all calls to the OpenGL API removed. However, instead of doing this instantaneously 40,000[1] per second, we compute an average power over one or more frames and subtract that average amount. Note that the energy consumption of the graphics driver is included.

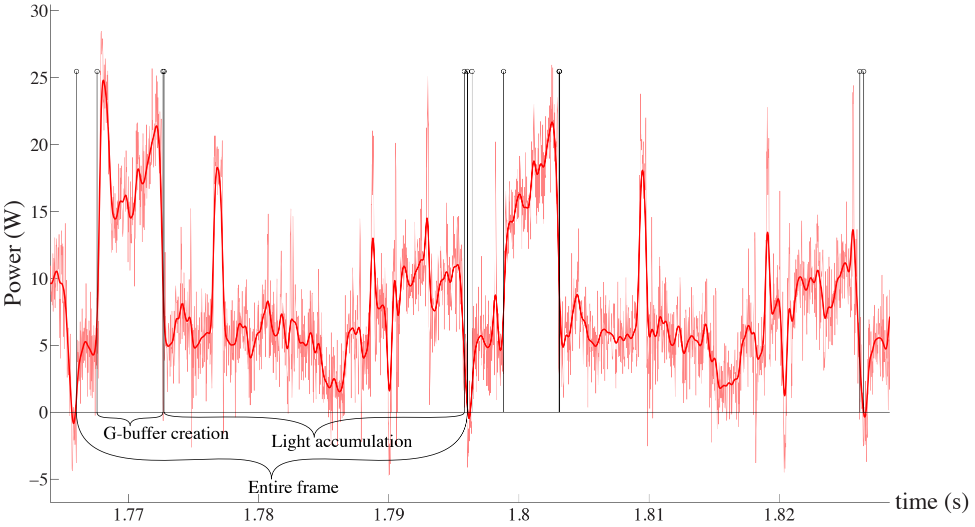

Figure 1. Measured power for deferred shading on an Ivy Bridge processor. The entire frame on the left uses 135 nJ/pixel. The G-buffer creation pass uses 47 nJ/pixel and the light-accumulation pass uses 84 nJ/pixel. Note that the removal of the CPU power pushes the curves below zero at times. This is because the idle power that is removed is an average over several complete frames, which reduces noise in the graphs (but still gives a correct average energy per pixel).

[1] Our power measurement station operates at 40 kHz

The power curves over time for two frames are shown in Figure 1. We estimated that the energy per pixel is 135 nJ/pixel including both rendering passes as well as any overhead in between. Moreover, we inserted time stamps into the rendering, which made it possible to isolate the G-buffer pass and the lighting-accumulation pass. The G-buffer pass uses 47 nJ/pixel, while the lighting accumulation pass uses 84 nJ/pixel. The rest of the rendering (overhead) uses $135-47-84=4$ nJ/pixel.

Let us convert these numbers to their inverses (performance per watt), i.e., $1/(135\cdot 10^{-9})=7.4$ Mpixels/joule for the full frame. Similarly, the G-buffer pass converts to 21.3 Mpixels/joule, and the lighting-accumulation pass to 11.9 Mpixels/joule. Reasoning about the overhead energy is very non-intuitive, since the numbers, 7.4, 21.3, and 11.9 Mpixels/joule, do not easily convey the energy overhead of the rendering. One could convert the overhead, i.e., 4 nJ/pixel, to its inverse, which becomes 250 Mpixels/joule, but this clearly does not help either in understanding what the overhead is.

We conclude that it is clear that performance per watt (fps per watt, frames per joule, pixels per joule, etc) should be avoided in both academic and industry research when examining energy/power efficiency.

Acknowledgements. Thanks to Aaron Lefohn, Charles Lingle, and Tom Piazza at Intel; thanks, in particular, to Jim Nilsson, Robert Toth, and Petrik Clarberg for feedback on the paper and also to the rest of the Advanced Rendering Technology team. We acknowledge support from the Swedish Research Council; Tomas Akenine-Möller is a *Royal Swedish Academy of Sciences Research Fellow* supported by a grant from the Knut and Alice Wallenberg Foundation.

References

AKENINE-MÖLLER, T., AND STRÖM, J. 2003. Graphics for the Masses: A Hardware Rasterization Architecture for Mobile Phones. *ACM Transactions on Graphics, 22*, 3, 801–808. 37

JOHNSSON, B., GANESTAM, P., DOGGETT, M., AND AKENINE-MÖLLER, T. 2012. Power Efficiency for Software Algorithms running on Graphics Processors. In *High Performance Graphics*, 67–75. 37, 38, 39

NVIDIA. 2012. GeForce GTX 680. Tech. rep. 37

Author Contact Information

Tomas Akenine-Möller
Department of Computer Science
Lund University
Box 118, 221 00 Lund, Sweden
tam@cs.lth.se

Björn Johnsson
Department of Computer Science
Lund University
Box 118, 221 00 Lund, Sweden
bjorn@cs.lth.se

Akenine-Möller and Johnsson, Performance per What?, *Journal of Computer Graphics Techniques (JCGT)* 1, 1, 37–41, 2012
http://jcgt.org/published/0001/01/03/

Received: 2012 Sept. 5
Recommended: 2012 Sept. 18
Published: 2012 Oct. 18

Corresponding Editor: Peter Shirley
Editor-in-Chief: Morgan McGuire