GPU – do you?

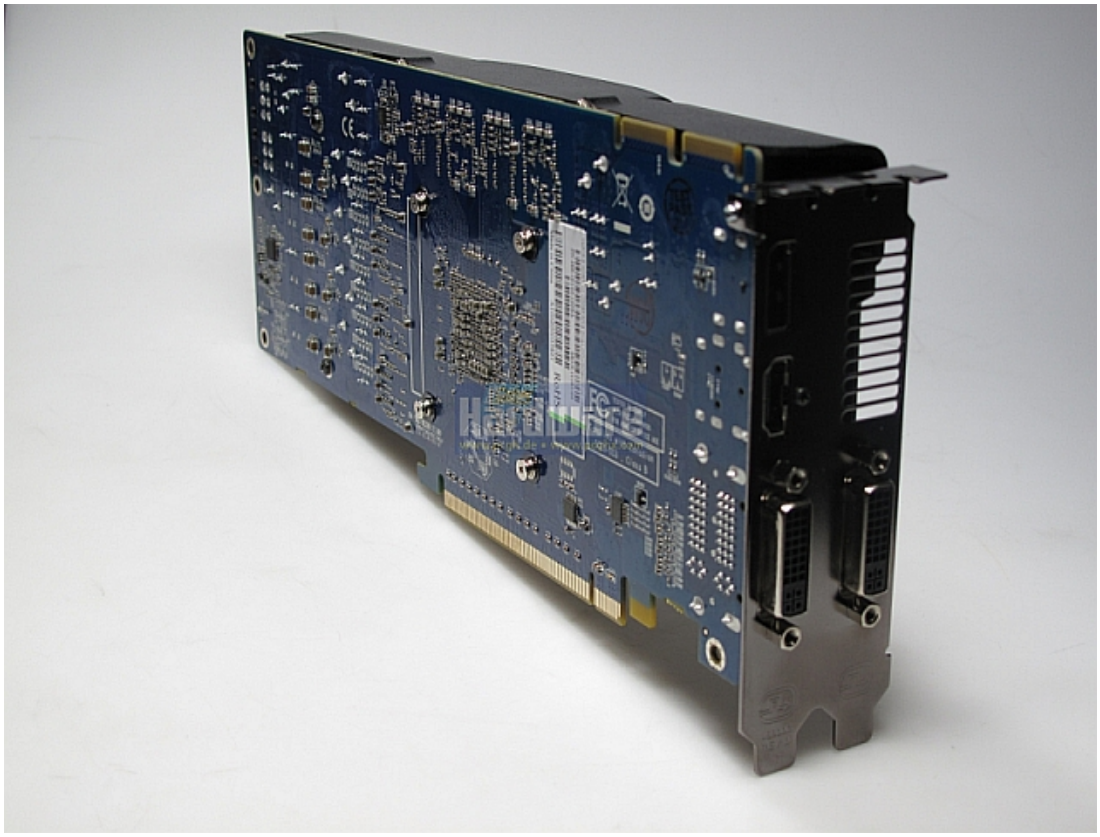
Jon Sonntag (Collatz) helps you find out – Part 1

What's all the fuss about GPU (video card) computing, and is it worth the effort? Do you still use exclusively your CPU for Distributed Computing and BOINC? Did you consider using your video card or do you already have some experience in GPU computing and consider doing upgrades? Is your old video card still helpful for computing? Jon Sonntag, project developer and administrator of Collatz Conjecture, helps you find out.

Secrets of performance: like a group of high school girls going to the bathtub

Hi Jon, Collatz has been one of the first projects focusing strongly on GPU computing. Now, if I was just starting BOINC with a decent computer having a medium or low end GPU, would you advise me to do GPU computing on it?

Absolutely. Even the slowest GPU can match the speed of a CPU. While Collatz was begun with GPU computing in mind, there are applications for CPUs as well.



PCB view on one of the most successful GPU computing video cards, the Radeon HD 5870.
Source: PC Games Hardware

How is it possible that a small GPU can outperform almost any brand new high end CPU?

GPUs calculate the colors for every pixel on the screen. They also calculate how to smooth the edges images, fade from one color to another, etc. To do that, GPUs have hundreds or even thousands of very small processors. GPUs can't do all the things a CPU can do such as communicate with memory, hard drives, the motherboard, etc. Like a group of high school girls going to the bathroom, GPU processors all have to do the same thing at the same time, otherwise known as parallel processing.

If one GPU processor needs to do something different, all the other processors have to wait until it is done. Lastly, the calculations can't be too complicated or take too long to complete. So, not every problem to be solved fits those criteria. For the ones that do, they benefit from having hundreds of GPU processors instead of only a few CPU cores to do the calculations.

For CPUs still Moore's law is valid that says computing power doubles every second year. For years now GPUs seemed growing way beyond this. How is this possible?

It all depends upon how you look at it. My AMD 4000+ has a faster clock speed than my Intel i7 950. But, my i7 can run 8 workunits at a time because it has 4 hyperthreaded cores. GPU processors haven't really increased that much in clock speed. The manufacturers have just packed more processors into the same amount of space. For example, a Radeon HD 3870 X2 runs at 825 MHz, has 1332 stream processors and has 1 teraFLOP of computing power. Since it is 4 years old, double the teraFLOPS twice and you have 4 teraFLOPS expected from a top of the line GPU today. Do we? Yes. A Radeon HD 7970 runs at 1Ghz (20% faster) but uses 2048 stream processors and generates 4.3 TeraFLOPS of computing power.

Note: I used a 3870 X2 in the example because it is about the same physical size as the dual slot HD 7970. The best part is that the 7970 requires 50W LESS power to run -- if you can afford the price tag.

One of the very first CUDA-capable video cards: a Geforce 8400 GS

Present and future: Tom Cruise and Dustin Hoffman in 'Rain Man'

Do you expect GPU computing to become more important over the next years?

For problems that can be solved using parallel processing, absolutely. In the past, it was very difficult to write software for GPUs. NVIDIA's CUDA toolkit made it easier. AMD's Brook+ never fully matured and AMD's CAL was too much like assembly language and difficult to learn. Now that BOINC supports OpenCL on GPUs, projects can write applications that will run on NVIDIA or AMD on OS X, Linux, and Windows with just minor changes.

How do you rate the quality and diversity of projects offering GPU computing today?

GPU computing has caught on the last couple years, but it has a way to go. At last count, only 25% of BOINC projects support GPUs and several only support either NVIDIA or AMD but not both. The good news is that the projects are spread across multiple areas from space science to protein structure prediction. The quality varies. PrimeGrid's applications are always rock solid while others such as Donate have severe stability issues for some people.

POEM needs 6 CPU cores to keep a single HD 6970 at 88% load and yet they consider that to be a GPU application. I guess they failed mathematics. There are several others which are also more of a CPU hybrid than GPU application. I think we will see more CPU/GPU hybrid applications as the OpenCL framework matures and as more developers learn to use it. Hopefully, those who have written some of the hybrid applications will improve them to where they are truly GPU applications.

Early ATI Stream card: an ASUS Radeon 2400 pro
Source: ozone3d.net

But there are projects who don't offer GPU apps. Couldn't they work much faster if they did?

Most would gain some benefit, but many of the algorithms are too complex or contain too many conditional operations or loops of an unknown size to run really well on a GPU. Einstein and POEM's applications are examples of what doesn't work as well. While they can do GPU calculations, they don't fully utilize the GPU and require a 100% of a CPU core in addition. I would consider those CPU applications that use part of a GPU. They do work faster though, so there is benefit to the project to offload some of the computations to the GPU.

I like to think of utilizing a PC's CPU and GPU resources much like how a business utilizes its employee resources. The database administrator should handle all the database tasks and the network administrator should handle the networking tasks. If there are lots of database tasks to do and no networking tasks, the network administrator may be able to help with the database tasks but won't be as fast or efficient as the database administrator.

Likewise, GPUs have certain types of computations at which they excel while CPUs have other types of computations at which they excel. If your GPU isn't being used at all, using 40% of it to help with a CPU task is of benefit. But, if you have both types of computations on which to work, I prefer to have the GPU work on computations where it excels and have the CPU work on computations where it excels. Otherwise, it is a waste of resources.

GPU vs CPU: who is better... Well, at what?

Will we all be just using our GPUs one day?

No. At least not in their present form. If you remember the movie Rain Man starring Tom Cruise and Dustin Hoffman, a GPU would be like Dustin Hoffman's character Raymond in that movie, an autistic savant, which can do one thing really really well. A CPU is more like Tom Cruise's character Charlie who is, compared to his brother, a jack of all trades. GPUs can't read data from hard drives, do multiple tasks at once, etc.

Companies are creating OpenCL computing devices for specific types of computations and as OpenCL matures, you will see a greater number of computations being offloaded to the GPU, but they won't ever replace a CPU.

Jon, thank you for taking your time and helping find out. I think you may have inspired some long-term crunchers to try what they can do on their GPU, no matter if old or brand new.

This was the first part of the interview with Jon Sonntag. In the second part we will talk about old video cards, laptops and the true reason behind his nickname Slicker. And finally Jon will give you a number of helpful tips on how to get started.

GPU – do you?

Jon Sonntag (Collatz) helps you find out – Part 2

What's all the fuss about GPU (video card) computing, and is it worth the effort? Do you still use exclusively your CPU for Distributed Computing and BOINC? Did you consider using your video card or do you already have some experience in GPU computing and consider doing upgrades? Is your old video card still helpful for computing? Jon Sonntag, project developer and administrator of Collatz Conjecture, helps you find out.

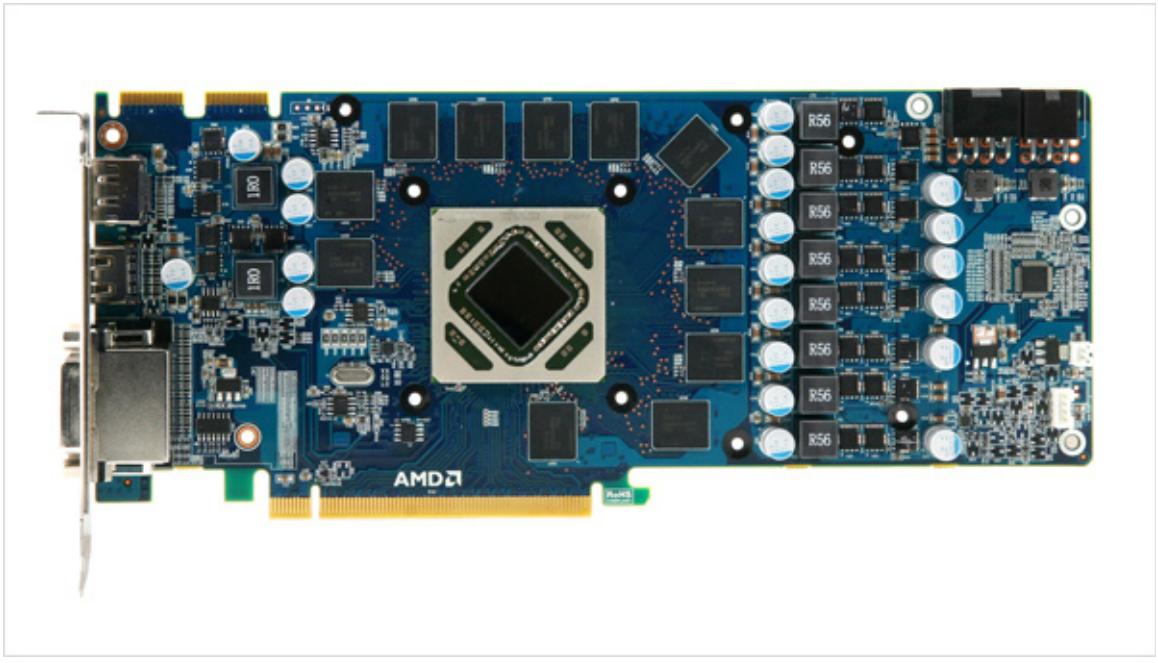
Do or don't? About monsters, laptops and old mules

Is there anyone who definitely should not use his GPU, for whatever reason?

1. Laptops are made for portability and because of that, there are trade offs in the amount of cooling and the number of calculations their GPUs can do. If the temperatures are too high (check out GPU-Z) one can either try underclocking or not use the GPU for computing. There are some aftermarket laptop coolers available which blow cool air onto the bottom of the laptop which, if the laptop uses the case as a heat sink, may be cool it enough.

2. Computers that are not your own and/or you do not have written permission to use. A verbal agreement is not enough. Running BOINC, whether on a CPU or a GPU uses electricity and causes wear and tear on a system as it is running 24x7 instead of only 8 hours a day.

3. Anyone from team Sicituradastra. ;-)



PCB of an AMD 7970.Source: overclock.net

Regarding crunching on laptops, you would consider TThrottle an option for mobile GPUs?

If the laptop fan can't keep the temperatures at a reasonable level, BOINC has settings to throttle the CPU but it works like a pulse-width modulator (e.g. a setting of 75% means it will run at 100% for 3 seconds and then 0% for 1 second). BOINC's doesn't have any options for the GPUs and the on-off method it uses for the CPUs isn't the best. TThrottle [http://www.efmer.eu/boinc/] actually monitors the temperatures and throttles back only when it needs to. It is a much better solution than the one provided in the BOINC Manager.

Worth the money? A brand new NVIDIA GK 104 chip
Source: legitreviews.com

Now, if I can do so much with so little effort, would you say I should buy a monster GPU just for science? Or can I achieve something with the hardware I have at home?

That really depends upon how much money you have and how much you love volunteer computing. If you can afford a monster GPU and the electricity to run it, go for it. You will be contributing anywhere from 10 to 100 times the work that you do now. But, since a monster GPU can easily cost as much as an entire PC and since most people don't have $1000 to spend on a GTX 690 or have other priorities (pay the rent or buy a GPU.....hmmm.....decisions, decisions.....)

Luckily, there is no need to purchase a new GPU. Almost any GPU made in the last several years can be used for computing. Even the slowest GPU will often match the output of the CPU and double the amount of work that the computer contributes to a project.

About Slicker, Indiana Jones and about the guy behind Collatz...

We 'talked' a lot about GPUs, so finally I'd like to ask some questions about you. I reckon I should ask you who Slicker is, if that's not too personal?

Like Indiana Jones, who took the name Indy from his dog, Slicker was our pure bred Weimaraner. His bout with cancer that had gone on for several years ended this past summer.

Slicker chasing squirrels

Jon, do you work in IT or mathematics, just thinking of your BOINC project?

Once upon a time, I was an elementary school teacher. I planned to get my master's degree and teach high school computer science. In Illinois, that meant getting state certified to teach high school mathematics because at the time, Illinois lumped computer science into the mathematics department. I taught a few IT courses at the college level while working at the university's computer center, but after a couple years, worked strictly in IT. That was twenty years ago and I've been in IT/IT Management ever since.

How did you come to the idea developing and running a project like Collatz?

It was a combination of things. Several BOINC projects supported NVIDIA GPUs but only MilkyWay supported AMD GPUs and that's not because of the project administrator, but because on a single volunteer (Crystal Physik) who wrote the AMD application for them. Unfortunately, the MilkyWay server couldn't keep up with AMD GPUs. I owned 4 of them and they would often be idle because there was no work for them.

At the same time, I worked at a publishing company which scored and reported standardized tests. I

proposed using BOINC to manage the scoring of the tests and using a GPU application to do the actual scoring computations would have greatly reduced the processing time considerably. So, I started looking for a simple problem that I could use to learn GPU programming and set up a prototype to convince my boss of the merits of distributed computing. Gipsel, as he is known on the Collatz Conjecture project, who was the volunteer from MilkyWay, really helped me out with the GPU applications and tutored me on the do's and don'ts. I had very little Linux experience, so Crunch3r helped with porting the applications to Linux.

Jon, would you do it again? And – will you eventually come up with a new GPU project?

The SETI.USA team was considering hosting a BOINC project as a team. I found a project and wrote applications for it. Since the team lost interest in running their own project, I may add it to the Collatz project as it is a mathematics based problem. Right now my focus is on sorting out some issues with the OpenCL applications. Later in Fall I expect to have more free time, so maybe I'll get around to adding it then.

How to speed up and how to get started

From your practical experience with Collatz, could you give us an example of the speed differences between crunching on GPUs vs crunching on CPUs?

HD 6970 vs Intel i7 950: The GPU is *132 times faster than a single i7 core*, and about *17 times faster than all 8 cores* combined. Many of the top crunchers have 2, 3, or even 4 GPUs in a single computer.

Is BOINC really up for supporting GPUs? Will I have to rack my head to make it work?

It is very, very easy to do. Install the latest video driver for your video card (a.k.a. GPU). Then join a BOINC project which has GPU applications for your type of GPU.

Take it easy: you don't need a triple-SLI for getting started. Source: Vizworld.com

Which projects would you recommend for getting started?

That depends upon your reasons for volunteer computing. If you have an area in which you are intrigued, find a project in that area that matches your type of GPU. If you don't know which type you have, the BOINC event log (Control+ Shift+E) will display the GPUs it recognizes when it first starts up. I believe Collatz supports the largest number and types of GPUs so if it can be used, it should work on Collatz.

If you are competitive and want to get the most cobblestones (a.k.a. BOINC credits) you can, POEM, Donate, DistrRTGen, and Moo!Wrapper pay the most credits per hour. Credits sometimes change, but your team's message board will likely have suggestions as to where your GPU can be best utilized.

Last but not least: What are your ultimate three tips for anyone who wants to start crunching on his GPU?

1. If using an aftermarket GPU, make sure the power supply in the PC is powerful enough to handle the extra load. Most GPUs' specifications will list the minimum size of the power supply in watts. Make sure yours is 20-30% higher than the minimum. If you need to upgrade the power supply, look for an 80-plus efficient rated one. While it will cost a little more, it will pay for itself by using less electricity. Most computers with embedded video (e.g. a VGA or HDMI jack right by the keyboard, USB, etc. jacks rather than on a separate card) will not be able to be used. A separate GPU can be added for as little as $15 that will crunch.

2. Keep it clean and cool. Your GPU will last longer and run cooler if you blow the dust off it every once or twice every year. Several of my GPUs are over 5 years old.

3. If you are having trouble, ask for help on the project's or your team's message boards. Make sure to give as much information as possible (project, type of GPU, video driver version, BOINC version, and specifically what isn't working). You will most likely get a reply within a day or two and be up and running in no time at all.

Jon, thank you – once more - for taking your time and helping find out. I think you may have inspired some long-term crunchers to try what they can do on their GPU, no matter if old or brand new.

The complete interview with Jon Sonntag will be available as blog and as pdf download.