Green Computing using Graphical Processing Units

Y.Navneeth Krishnan
1st Sem CSE,RVCE
8th mile mysore road
Bangalore-560059

Vipin Dwivedi
1st sem CSE,RVCE
8th mile mysore road
Bangalore 560059

Chandan N bhagwat
1st sem CSE, RVCE
8th mile mysore road
Bangalore 560059

ABSTRACT
Green computing is the process of reducing the power consumed by a computer and thereby reducing carbon emissions. The total power consumed by the computer excluding the monitor at its fully computative load is equal to the sum of the power consumed by the GPU in its idle state and the CPU at its full state .In our paper we have tried using the high processing speed of the GPU's to do the computational intensive parts while the sequential parts like storing data is made by the CPU.The GPU has 30-50 times more processing speed than the CPU .The GPU therefore does the 100% of the CPU work in its idle state .Hence the power consumed by the GPU will be low. Also when the GPU is doing all the work the CPU will remain at a load less than its idle load. Hence the power consumed will be equal to the power consumed by the CPU at a load less than its idle load plus the power consumed by a GPU.

General Terms
Green computing

Keywords
Green computing, CPU, GPU, power.

1. INTRODUCTION
Green computing refers to those practices adopted in the IT industry so as to reduce carbon emissions [1]. The practices may be both in hardware or software terms. The hardware obtained these days are usually very efficient if used properly .So there is more research in the coding part than in hardware be it with making the webpage of websites black so as to reduce power consumption to making shorter codes ,it all accounts for green computing [2-4].
In our paper we have used Graphical processing units for reducing the carbon emissions. Graphical processing units are the processing units which are used to calculate which pixel is to be lit up at what instant of time. Since these calculations have to be very fast so that the user doesn't feel any time delay in getting the required graphical output on the screen the processing speed and thereby the processing capacity of the graphical processing unit is very high. It is 30 -50 times higher than the processing speed of the CPU.
The reason why the GPU has higher processing speed than the CPU is that the CPU has other functions other than processing like storing of data, retrieving of data from the cache memory, primary memory, secondary memory, monitoring the other parts of the system along with many other functions. Whereas the GPU has only one job that is to perform calculations. Thereby due to a singularity function the processing speed of the GPU is very high. Presently GPU's are used for game physics calculations so as to provide a very interactive environment to the gaming user.

2. POWER CONSUMED BY A CPU
A CPU which has a processor at a higher clock speed will always consume more power .The maximum powerconsumed by the CPU also known as the thermal designpower. The thermal design power is given by P = Cv^2fWhere f is the frequency of the processor .The powerconsumed by the processor and hence the power consumed by the CPU can be reduced by a process called clock gating [5].

3. POWER CONSUMED BY A GPU
A GPU consumes much more power than a CPU as itsclock speed is very high, but the amount of watt consumed per unit Ghz clock speed is low when compared to a CPU .Therefore the CPU at its idle state consumes more power than a GPU in its idle state [6].

4. COMPARISON BETWEEN POWER CONSUMED BY A CPU AND GPU
The Tables below shows the power consumption by aCPU and GPU for various models.

Table 1: Power consumed by GPU

SL.NO	GPU USED	POWER INIDLE STATE (IN W)	POWER IN PEAK 2D(IN W)	POWER IN TYP 3D(IN W)	POWER IN PEAK 3D
1.	ATI RADEON X1900XT	26	45	75	-
2.	ATI RADEONHD 6990 CF(4)	43	0	577	-
3.	ATI RADEON HD 6950 OC(900/1440)(32)	23	64	182	223
4.	ATI RADEON 6950 1GB OC(900/1300)(4)	36	-	-	215
5.	ATI RADEON 6970 3x CF(6)	55	552	-	-

Table 2: Power consumed by a CPU in idle and full load states in watts

Processor name	Idle State	100% Load
Athlon64FX55 2.4GHz	150	194
Athlon64@2.80GHz	125	199
Pentium4XE@2.30GHz	206	318
Intelcorei3@2.40GHz	32	65
Intelcorei5@3.10Ghz	38	95
Intelcore i7@3.60GHz	34	95

Considering that we have an Intel Pentium 4 processor with and Raedon X1900 XT graphics card. From the table2 we find that the peak power consumed by a CPU for the computer (At max load to CPU) will be equal to the power consumed by the CPU and the GPU (idle state) that is equal to 318+26=344Watts. Also the speed at which the data is computed is also high in a GPU when compared to a CPU.

5. REDUCING POWER CONSUMPTION IN A DATA CENTRE USING GPU

In a datacenter a large amount of computational work is done using multiple processors mounted on different or on the same CPU. This generates a lot of heat. In our method the processor in the CPU is freed from the computationally intensive work. Now the CPU remains in its IDLE state. Since the processing capacity in the CPU is completely free all the power supplied to it will be dissipated as heat. Now using a process called clockgating we reduce the processing capacity of the CPU and thereby now the CPU consumes power less than its idlestate power. Now in the computer we have considered the CPU consumes power less than 206W when no load is supplied to it. Now the computationally intensive part is sent to the GPU. But this load does not constitute 100% of the load of the GPU as it did to the CPU it constitutes only 0.1% to 1% because the processing speed of the GPU is very high when compared to the CPU. Therefore 100% load in the CPU is 0.1 % load in the GPU. Hence the GPU will be more or less in its idle state.Now in the system the total power consumed will be less than 232W as against the 344 watts when the CPU is used for calculations .Thereby reducing 112 watts.

6. EXPERIMENTAL ANALYSIS

We have implemented the usage of GPU on an opencl platform. For the computationally intensive work we have used 2 matrices of 1024 and 2048 order respectively. The computer used contained an i5 processor with ATiraedon graphic card .The values of the matrices are randomly generated and stored and then the matrices are squared. The matrices are first squared using normal C coding and then squared on an OPENCL platform. The time required for both cases is found out. Also the total energy consumed when the open cl code is run is also found.

6.1. Coding in OPENCL

The general procedure for coding matrix multiplication on opencl is as follows [4].

Here we compute the following relation among matrices C = AB where

Dimensions of matrix A = (y, x) {width0, height0}.

Dimensions of matrix B = (z, y) {width1, width0}.

This results in a matrix C, with dimension = {z, x}.

For implementation on GPU we are using Radeon HD™ 4XXX series (7xx series GPUs), which does not use local memory.

Following are the steps required for matrixmultiplication

Step 1: Vectorize input Matrices, this reduces their width by a factor of 4.

Step 2: Matrix A is cached into local memory block.

Step 3: Calculate the required global threads = (width C / 4, height C / 4).

Step 4: Create a loop that runs for number of blocks of A in horizontal direction.

Step 5: Calculate global ids of threads from the particular block to load from matrix A depending on the index.

Step 6: Calculate global ids of threads from the particular block to load from matrix B depending on the index.

Step 7: Create a loop that runs for number of threads in horizontal direction in the block of A.

Step 8: Write the values to Matrix C.

6.1.1 Time analysis

When a matrix of order 1024 and 2048 containingrandom values is squared the time taken in open clcompiler as well as on C compiler is found out and it isdepicted in the table 3.

Table 3.Time taken for matrix multiplication on a CPU as well as GPU

Device	Matrix A	Matrix B	Time (sec)	Kernel Time (sec)
CPU	1024	1024	2.220	1.908
CPU	2016	2016	6.804	6.514
GPU	1024	1024	1.529	0.143
GPU	2016	2016	2.063	0.639

6.1.2 RESULTS

Screenshots of the results obtained when 1024 ordermatrices are multiplied on CPU and GPU is shown below.

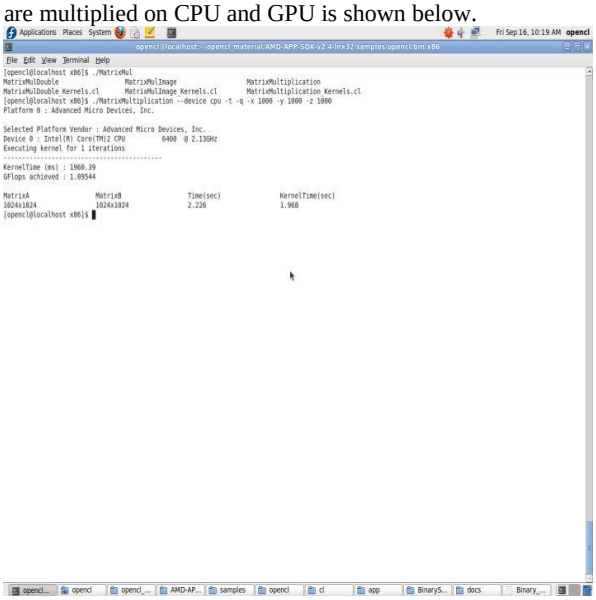

Fig 1: Screenshot showing the output of matrix multiplication of order 1024 on CPU

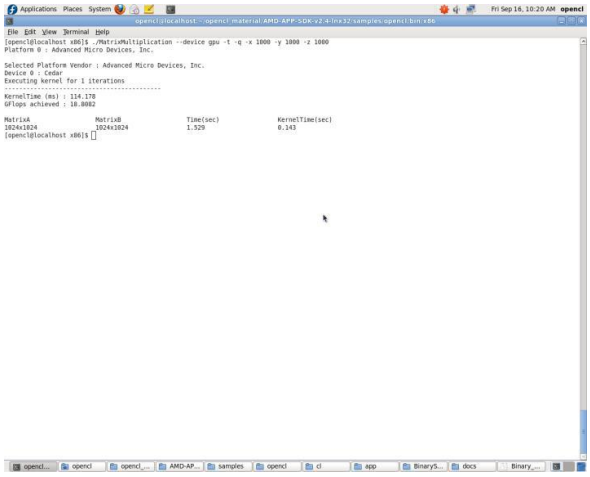

Fig 2: Screenshot showing the output of matrix of 1024 matrix

6.1.3 Outputs (POWER savings)

Using Windows gadgets we measure the core capacity which is used. The GPU percentage which had been used is also found out. There by coming to know which was used at what time during the execution. Also the power usage of processor is measured using the Lenovo battery meter.The power savings and the amount of clock speed used is depicted in the following screenshots.

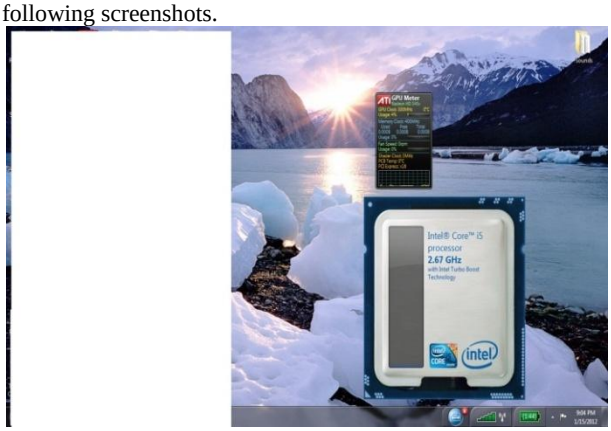

Fig3- During the Multiplication process the CPU is in its idle state .The CPU has 0 usage of clock speed

While the GPU is showing 4% usage of clockspeed

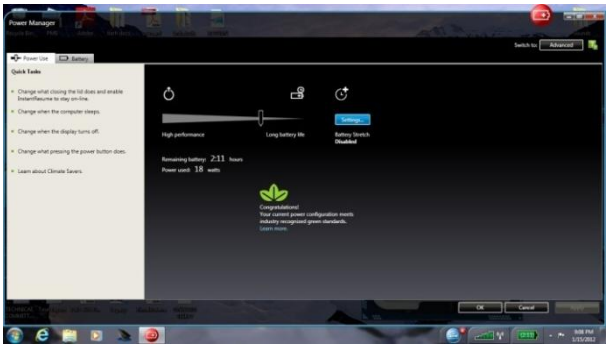

Fig 4:The power consumed using Lenovo battery meter (for processors) during the process

7. ACKNOWLEDGMENTS

Our thanks to Dr. G. Shobha, professor, CSE Department, R.V. College of Engineering who has contributed towards this work..

8. REFERENCES

[1] NVIDIA Corp. CUDA CUFFT Library, Version 1.1. 2007.

[2] K. Fatahalian, J. Sugerman,,& P. Hanraham, Understanding the efficiency of GPU Algorithm for Matrix-Matrix Multiplication.

[3] NIVIDIA Corp. CUDA Compute Unified Device Architecture. Programming Guide, Version 2.0, 2008

[4]NVIDIA Corp . GPU programming guide

[5] V. Garcia and E. Debreuve and M. Barlaud. Fast k nearest neighbor search using GPU. In Proceedings of the CVPR Workshop on Computer Vision on GPU, Anchorage, Alaska, USA, June 2008.

[6]www.GPGPU.org