deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Jetson Nano

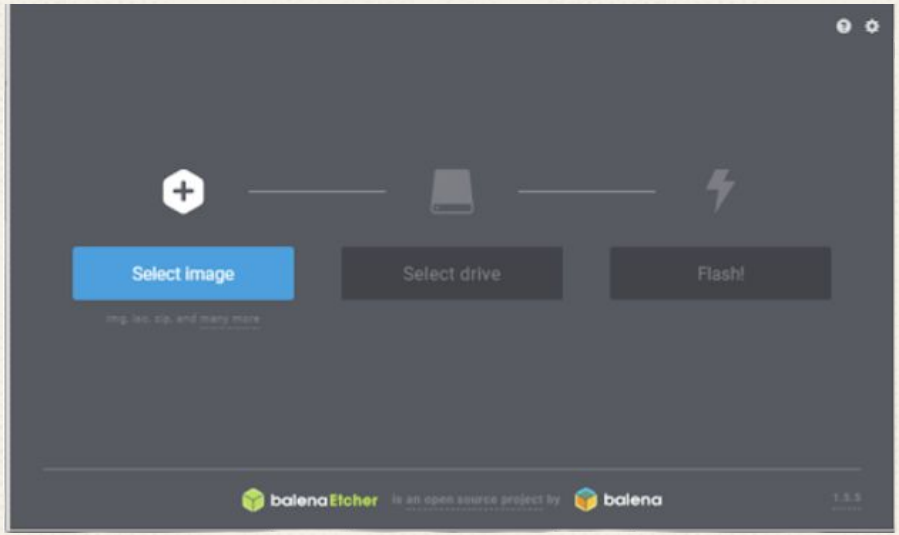
Project - Compare CPU and GPU

TP1 - Grupo 7
Lúcia Sousa 93086
Raquel Pinto 92948

14/06/2022 ASE - Arquitetura de Sistemas Embutidos

Etcher Imager

Etcher Imager is a way to install Linux4Tegra (Ubuntu 18.04). Is the board support package for Jetson (NVIDIA JetPack SDK).

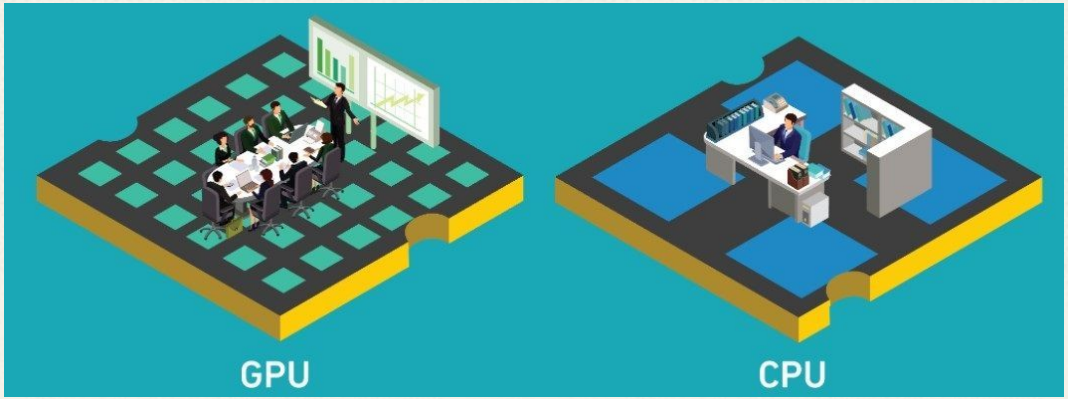


Access to Jetson

- Connect to a monitor via HDMI.
- Connect to a mouse and keyboard via USB.
- Connect the Wi-Fi adapter via USB.
- Connect the power input to port Micro-USB.

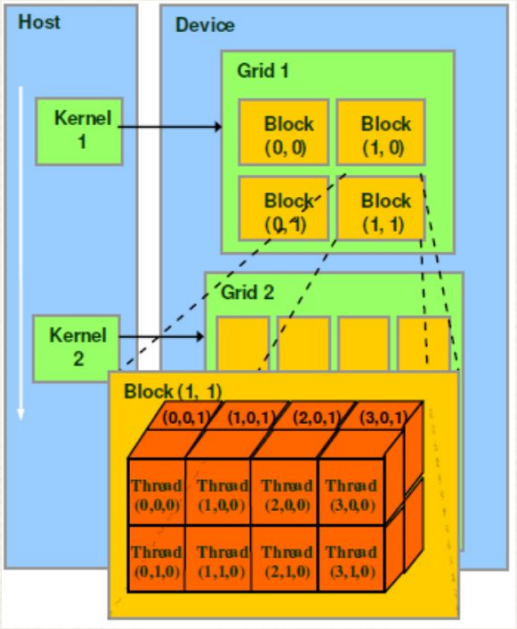
Project Idea

- For an operation, compare CPU performance and GPU performance.
- C = A^2 + B^2.
- A, B and C are arrays.
- Size is 1 048 576 = 1024*1024 (N).
- Calculates C 10000 (M) times.



Project Implementation

- Documents
  - project
    - operation.cu

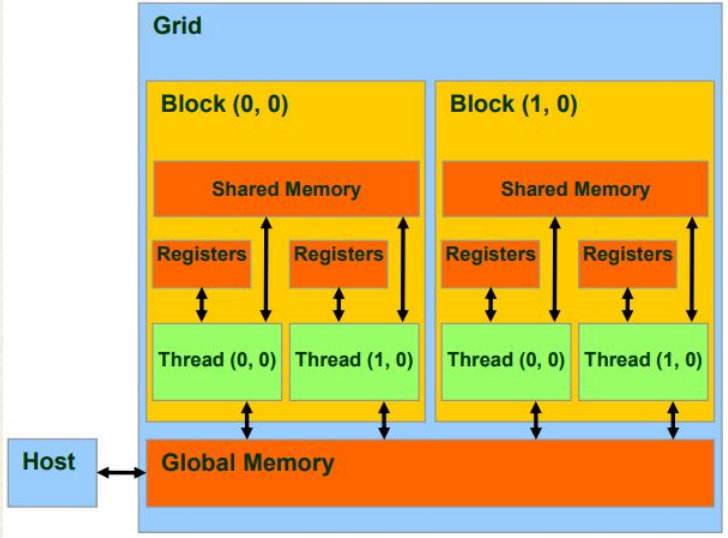


operation.cu

```
void serial_add(double *a, double *b, double *c, int n, int m){
        for(int index=0;index<n;index++) {
                for(int j=0;j<m;j++)  {
                        c[index] = a[index]*a[index] + b[index]*b[index];
                }
        }
}
// __global__ defines kernel function
__global__ void vector_add(double *a, double *b, double *c){
        int index = blockIdx.x * blockDim.x + threadIdx.x;
        for(int j=0;j<M;j++)  {
                c[index] = a[index]*a[index] + b[index]*b[index];
        }
}
```

Project Implementation

```
int main(){
        // For CPU
        start = clock();
        serial_add(a, b, c, N, M);
        end = clock();
        // For GPU with CUDA
        start = clock();
         // It's equal for others arrays
        cudaMalloc( (void **) &d_a, size );
        // THREADS_PER_BLOCK = 1024
        //myKernel<<< B, T >>>(arg1, ... );  B – 1D grid of that size (integer)
         T – 1D block of that size (integer)
        // NUMBER_BLOCKS = N + (THREADS_PER_BLOCK-1))/ THREADS_PER_BLOCK = 1024
        vector_add<<< (N + (THREADS_PER_BLOCK-1)) / THREADS_PER_BLOCK, THREADS_PER_BLOCK >>>( d_a, d_b, d_c );
        cudaFree( d_a ); // It's equal for others arrays
        end = clock();
}
```



Compilation, Execution and Demo

- Compile program nvcc operation.cu -o operation.
- Run program ./operation.

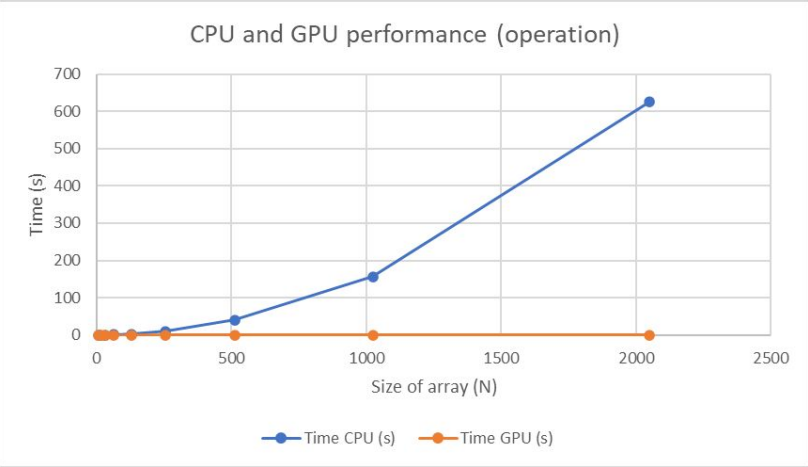
operation.cu

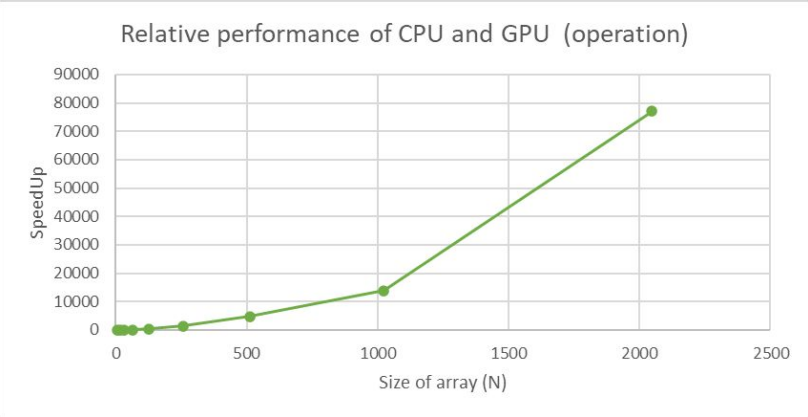
```
nvidia@nvidia-desktop:~/Documents/projeto$ ./operation
CPU: 156.109909 seconds
CUDA: 0.011253 seconds, Speedup: 13872.736328
```

palindrome.cu

```
nvidia@nvidia-desktop:~/Documents/projeto$ ./palindrome
CPU: 0.000003 seconds
CUDA: 0.007104 seconds, Speedup: 0.000422
```

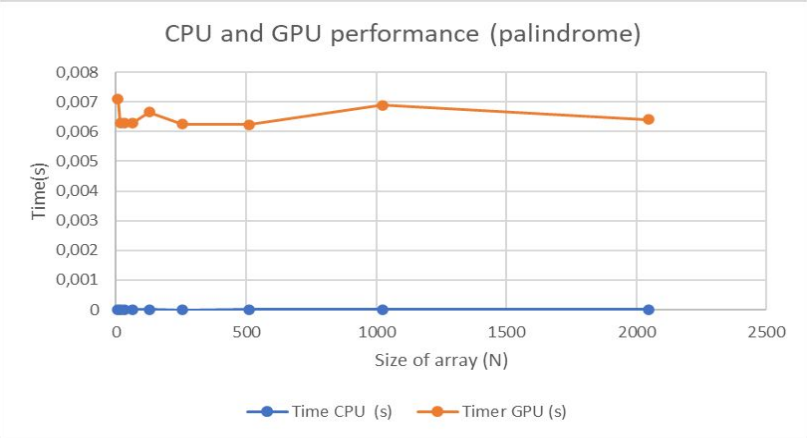
Conclusion (operation.cu)

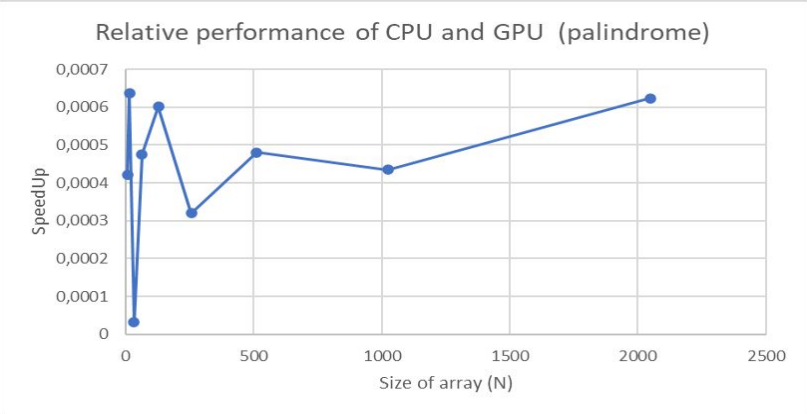




Size of array (N)	Time CPU (s)	Timer GPU (s)	speedUp
8	0.009524	0.0076	1.253158
16	0.038178	0.008227	4.640574
32	0.153977	0.006217	24.767092
64	0.615161	0.006262	98.080521
128	2.452388	0.006463	379.450409
256	9.84491	0.006758	1456.778687
512	39.793583	0.008429	4721.032227
1024	156.109909	0.011253	13872.73633
2048	624.711487	0.008104	77086.80469

Conclusion (palindrome.cu)





Size of array (N)	Time CPU (s)	Timer GPU (s)	speedUp
8	0.000003	0.007104	0.000422
16	0.000004	0.006285	0.000636
32	0.000002	0.006289	0.000318
64	0.000003	0.006296	0.000476
128	0.000004	0.006659	0.000601
256	0.000002	0.006246	0.00032
512	0.000003	0.006243	0.000481
1024	0.000003	0.006896	0.000435
2048	0.000004	0.006413	0.000624

Bibliography

- https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#intro
- https://github.com/LuigiMorelli/MoreWare
- https://docs.vmware.com/en/VMware-vSphere-Bitfusion/3.0/Example-Guide/GUID-ABB4A0B1-F26E-422E-85C5-BA9F2454363A.html
- https://jfrog.com/connect/post/installing-cuda-on-nvidia-jetson-nano/
- https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html?fbclid=IwAR2qENHQORIDS_JTy3IxNFsPNcQmXg3mPywbnwe45mDV1UzIPKQtf0VV0Sk#mandatory-post
- https://cs.calvin.edu/courses/cs/374/CUDA/CUDA-Thread-Indexing-Cheatsheet.pdf
- https://www3.nd.edu/~zxu2/acms60212-40212-S12/Lec-12-01.pdf
- http://15418.courses.cs.cmu.edu/fall2016content/lectures/07_gpuarch/07_gpuarch_slides.pdf