HPC_Assign2

October 27, 2020

[ ]:
```
!apt-get --purge remove cuda nvidia* libnvidia-*
!dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 dpkg --purge
!apt-get remove cuda-*
!apt autoremove
!apt-get update
```

```
Reading package lists… Done
Building dependency tree
Reading state information… Done
Note, selecting 'nvidia-kernel-common-418-server' for glob 'nvidia*'
Note, selecting 'nvidia-325-updates' for glob 'nvidia*'
Note, selecting 'nvidia-346-updates' for glob 'nvidia*'
Note, selecting 'nvidia-driver-binary' for glob 'nvidia*'
Note, selecting 'nvidia-331-dev' for glob 'nvidia*'
Note, selecting 'nvidia-304-updates-dev' for glob 'nvidia*'
Note, selecting 'nvidia-compute-utils-418-server' for glob 'nvidia*'
Note, selecting 'nvidia-384-dev' for glob 'nvidia*'
Note, selecting 'nvidia-libopencl1-346-updates' for glob 'nvidia*'
Note, selecting 'nvidia-driver-440-server' for glob 'nvidia*'
Note, selecting 'nvidia-340-updates-uvm' for glob 'nvidia*'
Note, selecting 'nvidia-dkms-450-server' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-source-440-server' for glob 'nvidia*'
Note, selecting 'nvidia-331-updates-uvm' for glob 'nvidia*'
Note, selecting 'nvidia-cg-toolkit' for glob 'nvidia*'
Note, selecting 'nvidia-dkms-440-server' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-390' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-410' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-415' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-418' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-430' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-435' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-440' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-450' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common-455' for glob 'nvidia*'
Note, selecting 'nvidia-opencl-icd-340-updates' for glob 'nvidia*'
Note, selecting 'nvidia-384-updates' for glob 'nvidia*'
Note, selecting 'nvidia-utils-440-server' for glob 'nvidia*'
```

```
[39.3 kB]
Fetched 11.2 MB in 2s (4,680 kB/s)
Reading package lists… Done
```

```
[ ]: !wget https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/
      ↪cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 -O␣
      ↪cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
     !dpkg -i cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
     !apt-key add /var/cuda-repo-9-2-local/7fa2af80.pub
     !apt-get update
     !apt-get install cuda-9.2
```

```
--2020-10-26 06:17:55--
https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-
ubuntu1604-9-2-local_9.2.88-1_amd64
Resolving developer.nvidia.com (developer.nvidia.com)… 152.199.0.24
Connecting to developer.nvidia.com (developer.nvidia.com)|152.199.0.24|:443…
connected.
HTTP request sent, awaiting response… 302 Found
Location: https://developer.download.nvidia.com/compute/cuda/9.2/secure/Prod/loc
al_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb?CvXVJSThQoKlPgeb
_BZlgCnJKXggcdbYlgNYfB4Pz0VZtgTqSzYXMgVlUOwb8FgwUTGCvfHkTHQeRsclYg9RQ-JNgk1hY-4o
9p9nDtGaMyNMNIR39T6h4CwiiIePaKS-U8of4Sv3IeNC_5umwxuAxcf1TrHqzJV5tw7WDsQq6tX_o1JE
6zuKJiP61BFzcKMnIH2-0xIs7HHx3dCRpwo [following]
--2020-10-26 06:17:55--  https://developer.download.nvidia.com/compute/cuda/9.2/
secure/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb?C
vXVJSThQoKlPgeb_BZlgCnJKXggcdbYlgNYfB4Pz0VZtgTqSzYXMgVlUOwb8FgwUTGCvfHkTHQeRsclY
g9RQ-JNgk1hY-4o9p9nDtGaMyNMNIR39T6h4CwiiIePaKS-U8of4Sv3IeNC_5umwxuAxcf1TrHqzJV5t
w7WDsQq6tX_o1JE6zuKJiP61BFzcKMnIH2-0xIs7HHx3dCRpwo
Resolving developer.download.nvidia.com (developer.download.nvidia.com)…
152.195.19.142
Connecting to developer.download.nvidia.com
(developer.download.nvidia.com)|152.195.19.142|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 1267391958 (1.2G) [application/x-deb]
Saving to: ‘cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb’

cuda-repo-ubuntu160 100%[===================>]   1.18G   257MB/s    in 4.7s

2020-10-26 06:18:00 (255 MB/s) - ‘cuda-repo-
ubuntu1604-9-2-local_9.2.88-1_amd64.deb’ saved [1267391958/1267391958]

Selecting previously unselected package cuda-repo-ubuntu1604-9-2-local.
(Reading database … 122374 files and directories currently installed.)
Preparing to unpack cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb …
Unpacking cuda-repo-ubuntu1604-9-2-local (9.2.88-1) …
Setting up cuda-repo-ubuntu1604-9-2-local (9.2.88-1) …
OK
```

```
Processing triggers for systemd (237-3ubuntu10.42) …
Processing triggers for man-db (2.8.3-2ubuntu0.1) …
Processing triggers for dbus (1.12.2-1ubuntu1.2) …
Processing triggers for hicolor-icon-theme (0.17-2) …
Processing triggers for fontconfig (2.12.6-0ubuntu2) …
Processing triggers for mime-support (3.60ubuntu1) …
Processing triggers for libc-bin (2.27-3ubuntu1.2) …
/sbin/ldconfig.real: /usr/local/lib/python3.6/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link
```

[ ]:
```
!nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Wed_Apr_11_23:16:29_CDT_2018
Cuda compilation tools, release 9.2, V9.2.88
```

[2]:
```
!pip install git+git://github.com/andreinechaev/nvcc4jupyter.git
```

```
Collecting git+git://github.com/andreinechaev/nvcc4jupyter.git
  Cloning git://github.com/andreinechaev/nvcc4jupyter.git to /tmp/pip-req-build-
jg87b4ms
  Running command git clone -q git://github.com/andreinechaev/nvcc4jupyter.git
/tmp/pip-req-build-jg87b4ms
Building wheels for collected packages: NVCCPlugin
  Building wheel for NVCCPlugin (setup.py) … done
  Created wheel for NVCCPlugin: filename=NVCCPlugin-0.0.2-cp36-none-any.whl
size=4307
sha256=d33f5e658b4abaa2fb85eeebacee8c98be96b91e9be3b6d54e638a9b85841513
  Stored in directory: /tmp/pip-ephem-wheel-cache-w2ptktdq/wheels/10/c2/05/ca241
da37bff77d60d31a9174f988109c61ba989e4d4650516
Successfully built NVCCPlugin
Installing collected packages: NVCCPlugin
Successfully installed NVCCPlugin-0.0.2
```

[3]:
```
%load_ext nvcc_plugin
```

```
created output directory at /content/src
Out bin /content/result.out
```

[4]:
```
%%cu
#include <cstdio>
#include <cmath>
#include <iostream>

__global__ void sumi(double* a, double* b, double *c,int n)
{
```

```
    int id = blockIdx.x*blockDim.x+threadIdx.x;

    if (id<n)
      c[id]=a[id]+b[id];
}

int main()
{

    int n;
    n = 100000;
    double a[n],b[n],c[n];
    int size_arr=n;
    for (int i = 0; i < n; i++) {
        a[i] = sin(i)*sin(i);
        b[i] = cos(i)*cos(i);
        //std::cout<<a[i]<<"+"<<b[i]<<" ";
    }

    cudaEvent_t start, end;
    double *ad, *bd, *cd;
    size_t sizeA = n * sizeof(double);
    cudaMalloc(&ad,sizeA);
    cudaMemcpy(ad,a,sizeA,cudaMemcpyHostToDevice);
    cudaMalloc(&bd,sizeA);
    cudaMemcpy(bd,b,sizeA,cudaMemcpyHostToDevice);
    cudaMalloc(&cd,sizeA);

    int blockSize = 256;
    int gridSize = (int)ceil((float)n/blockSize);

    cudaEventCreate(&start);
    cudaEventCreate(&end);
    cudaEventRecord(start);

    sumi<<<gridSize, blockSize>>>(ad, bd, cd, n);

    cudaEventRecord(end);
    cudaEventSynchronize(end);

    float time = 0;
    cudaEventElapsedTime(&time, start, end);


    cudaMemcpy(c,cd,sizeA,cudaMemcpyDeviceToHost);

    double sum=0;
```

```
    for(int i=0; i<n; i++) {
        sum+=c[i];
    }
    std::cout<<"Result parallel: "<<sum/size_arr<<"\n";
    std::cout<<"The time required for parallel: ";
    std::cout<<time;

    cudaEventRecord(start);
    sum=0;
    for(int i=0;i<size_arr;i++) {
        sum+=a[i]+b[i];
    }

    cudaEventRecord(end);
    cudaEventSynchronize(end);

    time = 0;
    cudaEventElapsedTime(&time, start, end);

    std::cout<<"\nResult : "<<sum/size_arr<<"\n";
    std::cout<<"The time required : ";
    std::cout<<time;


}
```

```
Result parallel: 1
The time required for parallel: 0.025728
Result : 1
The time required : 0.358592
```

[6]:
```
%%cu
#include <cstdio>
#include <cmath>
#include <iostream>

__global__ void sumi(double* a, double* b, double *c,int n)
{
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;
    if( row < n && col < n){
      double value = 0;
      for(int k = 0; k < n; k++){
        value += a[row * n + k] * b[k * n + col];
      }
      c[row * n + col] = value;
    }
```

```
}

int main()
{

    int n;
    n = 100;
    double a[n*n],b[n*n],c[n*n];
    int size_arr=n;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            a[i*n+j] = sin(i);
            b[i*n+j] = cos(j);
        }
    }

    cudaEvent_t start, end;
    double *ad, *bd, *cd;
    size_t sizeA = n * n * sizeof(double);
    cudaMalloc(&ad,sizeA);
    cudaMemcpy(ad,a,sizeA,cudaMemcpyHostToDevice);
    cudaMalloc(&bd,sizeA);
    cudaMemcpy(bd,b,sizeA,cudaMemcpyHostToDevice);
    cudaMalloc(&cd,sizeA);

    int BLOCK_SIZE = 256;
    dim3 dim_grid(ceilf(n/(float)BLOCK_SIZE), ceilf(n/(float)BLOCK_SIZE), 1);
    dim3 dim_block(BLOCK_SIZE, BLOCK_SIZE, 1);


    cudaEventCreate(&start);
    cudaEventCreate(&end);
    cudaEventRecord(start);

    sumi<<<dim_grid, dim_block>>>(ad, bd, cd, n);

    cudaEventRecord(end);
    cudaEventSynchronize(end);

    float time = 0;
    cudaEventElapsedTime(&time, start, end);


    cudaMemcpy(c,cd,sizeA,cudaMemcpyDeviceToHost);

    double sum=0;
    for(int i=0; i<n; i++) {
```

```
        sum+=c[i];
    }
    std::cout<<"Result parallel: "<<sum<<"\n";
    std::cout<<"The time required for parallel: ";
    std::cout<<time;

    cudaEventRecord(start);
    sum=0;
    double mat[size_arr*size_arr];
    for(int i = 0; i < size_arr; i++)
        for(int j = 0; j < size_arr; j++){
            double value = 0.0f;
            for(int k = 0; k < size_arr; k++){
                value += a[i * size_arr + k] * b[k * size_arr + j];
            }
            mat[i * size_arr + j] = value;
        }

    cudaEventRecord(end);
    cudaEventSynchronize(end);

    sum=0;
    for(int i=0; i<n; i++) {

        sum+=mat[i];
    }

    time = 0;
    cudaEventElapsedTime(&time, start, end);

    std::cout<<"\nResult : "<<sum<<"\n";
    std::cout<<"The time required : ";
    std::cout<<time;
}
```

```
Result parallel: 0
The time required for parallel: 0.002912
Result : 0
The time required : 3.48189
```

[42]:
```
%%cu
#include <cstdio>
#include <cmath>
#include <iostream>

__global__ void sumi(int* a, int* b, int *c,int n)
{
```

```
    int id = threadIdx.x + ( blockIdx.x * blockDim.x );
    int sum = 0;
    if(id < n){
        int start = id*n;
        for(int j=0; j<n; j++){
            sum += b[start + j] * a[j];
        }
        c[id] = sum;
    }
}

int main()
{

    int n;
    n = 1000;
    int a[n],b[n*n],c[n];
    int size_arr=n;
    for (int i = 0; i < n; i++) {
        a[i]=rand()%n;
        for (int j = 0; j < n; j++) {
            b[i*n+j] = rand()%n;
        }
    }

    cudaEvent_t start, end;
    int *ad, *bd, *cd;
    size_t sizeA = n *sizeof(int);
    cudaMalloc(&ad,sizeA);
    cudaMemcpy(ad,a,sizeA,cudaMemcpyHostToDevice);
    cudaMalloc(&bd,sizeA*n);
    cudaMemcpy(bd,b,sizeA*n,cudaMemcpyHostToDevice);
    cudaMalloc(&cd,sizeA);

    cudaEventCreate(&start);
    cudaEventCreate(&end);
    cudaEventRecord(start);

    sumi<<<n/256+1, 256>>>(ad, bd, cd, n);

    cudaEventRecord(end);
    cudaEventSynchronize(end);

    float time = 0;
    cudaEventElapsedTime(&time, start, end);
```

```
    cudaMemcpy(c,cd,sizeA,cudaMemcpyDeviceToHost);

    int sum=0;
    for(int i=0; i<n; i++) {
        sum+=c[i];
    }
    std::cout<<"Result parallel: "<<sum<<"\n";
    std::cout<<"The time required for parallel: ";
    std::cout<<time;

    cudaEventRecord(start);
    sum=0;
    int mat[size_arr];
    for(int i = 0; i < size_arr; i++){
        int value=0;
        for(int j = 0; j < size_arr; j++){
            value += a[j] * b[i * size_arr + j];
        }
        mat[i] = value;
    }

    cudaEventRecord(end);
    cudaEventSynchronize(end);

    sum=0;
    for(int i=0; i<n; i++) {
        sum+=mat[i];
    }

    time = 0;
    cudaEventElapsedTime(&time, start, end);

    std::cout<<"\nResult : "<<sum<<"\n";
    std::cout<<"The time required : ";
    std::cout<<time;
}
```

```
Result parallel: 397391173
The time required for parallel: 0.343264
Result : 397391173
The time required : 2.98192
```

[ ]: