#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include <cuda.h>

#include <math.h>

#include <omp\_kernel.cu>

#define row 250

#define col 256

#define id(m,n,ld) (((n)\*(ld)+(m)))

float sign(float x);

int compare\_float(float f1, float f2);

int s=90;

int main()

{int n,m,MN=row\*col,k,size=256,i,j,\*h\_m,h\_i[90],o,ii,il,l=0;

float h\_Phi[250][256],\*h\_x,a,dtime,x[256][1],\*h\_y,h\_r[250][1],h\_Phit[256][250],\*h\_u,\*h\_d,jj=250,h\_max[90];

float norm=sqrt(jj),\*d\_a,\*d\_b,\*d\_y,\*d\_Phit,\*h\_X,aa;

h\_y=(float\*)malloc(250\*sizeof(float));

h\_u=(float\*)malloc(250\*sizeof(float));

h\_d=(float\*)malloc(256\*sizeof(float));

cudaMalloc((void\*\*)&d\_a,row\*col\*sizeof(float));

cudaMalloc((void\*\*)&d\_b,col\*1\*sizeof(float));

cudaMalloc((void\*\*)&d\_y,row\*1\*sizeof(float));

cudaMalloc((void\*\*)&d\_Phit,col\*row\*sizeof(float));

dim3 dimGrid(20,20,1);

dim3 dimBlock(20,20,1);

srand(time(NULL));

clock\_t start=clock();

for(n=0;n<row;n++)

{

for(m=0;m<col;m++)

{

a=2.0\*rand()/(float)RAND\_MAX-1.0/norm;

h\_Phi[n][m]=a;

}

}

//printf("%f",h\_Phi[1][0]);

h\_X=(float\*)calloc(col,sizeof(float));

for(k=0;k<col;k++)

{

a=2.0\*rand()/(float)RAND\_MAX-1.0;

h\_X[rand()%col]=a;

}

for(i=0;i<256;i++)

{ x[i][0]=0;

}

for(i=1;i<=45;i++)

{

for(j=1;j<=i\*2;j++)

{x[j][0]=1;

}

cudaMemcpy(d\_a,h\_Phi,row\*col\*sizeof(float),cudaMemcpyHostToDevice);

cudaMemcpy(d\_b,x,col\*1\*sizeof(float),cudaMemcpyHostToDevice);

cudaEvent\_t start, stop;

cudaEventCreate(&start);

cudaEventCreate(&stop);

// Start record

cudaEventRecord(start, 0);

matr\_mul<<<dimGrid,dimBlock>>>(d\_a,d\_b,d\_y,250,256);

cudaMemcpy(h\_y,d\_y,250\*1\*sizeof(float),cudaMemcpyDeviceToHost);

//for(int m=0;m<250;m++)

//printf("%f\n",\*(h\_y+m));

for(ii=0;ii<250;ii++)

h\_r[ii][0]=h\_y[ii];

matr\_trans<<<dimGrid,dimBlock>>>(d\_a,d\_Phit,256,250);

//printf("%f",d\_Phit[1]);

matr\_mul<<<dimGrid,dimBlock>>>(d\_Phit,d\_b,d\_y,256,250);

cudaMemcpy(h\_u,d\_y,256\*1\*sizeof(float),cudaMemcpyDeviceToHost);

// printf("%f\n",h\_u[255]);

float max=h\_u[0];

for(j=1;j<256;j++)

{if(max<h\_u[j])

{ max=h\_u[j];

}

}

h\_max[l]=max;

for(int p=0;p<256;p++)

{//printf("%f,%f\n",h\_max[l],h\_u[255]);

if(compare\_float(h\_max[l],h\_u[p]))

{//printf("%d\n",p);

int m=p;

h\_i[l]=m;

}}

l++;

cudaEventRecord(stop, 0);

cudaEventSynchronize(stop);

float elapsedTime;

cudaEventElapsedTime(&elapsedTime, start, stop); // that's our time!

// Clean up:

cudaEventDestroy(start);

cudaEventDestroy(stop);

// printf("x");

aa=elapsedTime;

}

for(i=0;i<45;i++)

{printf("Index=%d\n",h\_i[i]);

printf("Value=%f\n",h\_max[i]);

}

printf("Elapsed time : %.2f min\n" ,(aa/100000)/10);

cudaFree(d\_a);

cudaFree(d\_b);

cudaFree(d\_y);

cudaFree(d\_Phit);

}

float sign(float x)

{return (x>=0)-(x<0);}

int compare\_float(float f1, float f2)

{

float precision = 0.000001;

if (((f1 - precision) < f2) &&

((f1 + precision) > f2))

{

return 1;

}

else

{

return 0;

}

}