
High Performance Computing

Departamento de Ingeniería en Informática

PEP1

1. (1.5) Sea A un vector de n enteros. Usando OpenMP, escriba un código paralelo que calcule la media de A , es decir

$$m = \frac{1}{n} \sum_{i=0}^{n-1} A_i$$

Asuma que la memoria para A ya fue procurada y los datos fueron leídos.

2. Las siguientes líneas de códigos corresponden a la invocación de un kernel que traspone una matriz cuadrada:

```
dim3 blocksize;
dim3 gridsize;
gridsize.x = gridsize.y = N/512;
blocksize.x = blocksize.y = 512;
transpose<<<gridsize, blocksize>>>(A, B, tmp, N);
```

es decir $B = A^t$. Asuma que la memoria para A , B y `tmp` ya ha sido procurada, tanto en host como en GPU, como arreglos unidimensionales de $N \times N$ flotantes. `tmp` es memoria adicional usada como buffer en el kernel. Asuma que N es múltiplo de 512.

- (a) (0.5) Escriba un kernel en C y CUDA que trasponga la matriz.
(b) (1.0) Escriba el kernel, asumiendo ahora que no hay memoria adicional, es decir

```
transpose<<<gridsize, blocksize>>>(A, B, N);
```

3. (1.5) Explique cómo la *ocupancia* de un kernel podría estar (o no) asociada al speedup del código paralelo.
4. (1.5) Suponga que le han solicitado construir un código paralelo para implementar el adversario en un juego de dominó. Diga con argumentación cuál de las tecnologías vistas en clases sería la más apropiada.