Assignment #1: The Big Dot

The dot product of two vectors $a = (a_{0,}\ a_{1,}\ \ldots, a_{n-1})$ and $b = (b_{0,}\ b_{1,}\ \ldots,$ $b_{n-1})$, written $a \cdot b$, is simply the sum of the component-by-component products:

$$a \cdot b = \sum_{i=0}^{n-1} a_i \times b_i$$

Dot products are used extensively in computing and have a wide range of applications. For instance, in 3D graphics (n = 3), we often make use of the fact that $a \cdot b = |a||b|cos\theta$, where | | denotes vector length and θ is the angle between the two vectors.

In this assignment, you are expected to:

1. Write CUDA code to compute in parallel the dot product of two (possibly large N = 100,000, or N = 1024*1024) random single precision floating point vectors;
2. Write two functions to compute the results on the CPU and GPU, and compare the two results to check for correctness (1.0e-6);
    - float *CPU_big_dot(float *A, float *B, int N);
    - float *GPU_big_dot(float *A, float *B, int N);
3. Print performance statistics with timer function;
    - CPU: Tcpu = Total computation time for CPU_big_dot();
    - GPU: Tgpu = Total computation time for GPU_big_dot();
        - Memory allocation and data transfer from CPU to GPU time
        - Kernel execution time
        - Data transfer from GPU to CPU time
    - Speedup = CPU/GPU
4. Analyze the performance results in a few sentences.
    - Which one runs faster?

- What’s the reason for that? Problem size, overhead, etc.

Timer functions

```
#include <sys/time.h>
long long start_timer() {
    struct timeval tv;
    gettimeofday(&tv, NULL);
    return tv.tv_sec * 1000000 + tv.tv_usec;
}

long long stop_timer(long long start_time, char *name) {
    struct timeval tv;
    gettimeofday(&tv, NULL);
    long long end_time = tv.tv_sec * 1000000 + tv.tv_usec;
    Printf(“%s: %.5f sec\n”, name, ((float) (end_time – start_time)) /
    (1000 * 1000));
    return end_time – start_time;
}
```