

Forum:[Hack Forums](#)

Thread: [Cudafy and basic parallel for/foreach](#)

991z3r0 - November 26, 2018, 7:06 pm

So obviously doing parallel for and foreach in c# isn't that bad to do however i have to make some custom code that will process extremely faster if using CUDA cores/GPU.

I have a sample code to just try to understand but im drawing blanks on what/how to adjust the current code to test some simple foreach from a list of strings.

This is the code:

<https://pastebin.com/pPz6GawA>

So what im confused on is the whole VectorStruct[], is it literally like a list of work to be done and sent to the GPU?

I'm also confused as to how each of the for loops come into play for this example. i assume theKernal is the actual code being executed on the cores?

I guess any basic breakdown on how this works and how you're supposed to code a parallel for/foreach correctly using all cores.

deepshit - November 29, 2018, 3:47 am

VectorStruct is not list of works.It's just a vector.(math vector)

thekernel is the gpu thread and the vectors are simply its parameters.

first loop is initializing vectors and then in gpu thread they are added together.(vector a and b)

then it gets the result vector from gpu.(vector c)

I think you can use it like this :

```
gpu.Launch(1, N, ((Action<GThread>) thekernel));
```

```
public static void thekernel(GThread thread)
{
    //whatever code
}
```

you can pass anything instead of vectors.

991z3r0 - November 29, 2018, 10:58 am

(Nov 29, 2018 04:47 AM)deepshit Wrote: [\[CITE\]](#) VectorStruct is not list of works.It's just a vector.(math vector)

thekernel is the gpu thread and the vectors are simply its parameters.

first loop is initializing vectors and then in gpu thread they are added together.(vector a and b)

then it gets the result vector from gpu.(vector c)

I think you can use it like this :

```
gpu.Launch(1, N, ((Action<GThread>) thekernel));
```

```
public static void thekernel(GThread thread)
{
    //whatever code
}
```

}

you can pass anything instead of vectors.

Alright that makes more sense. So would this disperse the workload to all cores? Or is this kinda just submitting one "load" of work to do?

Or would the vector's itself be part of cudafy creating work and spreading to the cores? essentially i would want 100% or as many as possible cuda cores to process the work from a list of items if that makes sense

991z3r0 - December 4, 2018, 3:46 pm

(Nov 29, 2018 04:47 AM)deepshit Wrote: [\[CITE\]](#) VectorStruct is not list of works.It's just a vector.(math vector)

thekernel is the gpu thread and the vectors are simply its parameters.

first loop is initializing vectors and then in gpu thread they are added together.(vector a and b)

then it gets the result vector from gpu.(vector c)

I think you can use it like this :

```
gpu.Launch(1, N, ((Action<GThread>) thekernel));
```

```
public static void thekernel(GThread thread)
{
    //whatever code
}
```

you can pass anything instead of vectors.

Wait sorry to sound like a mega noob but when i try to use your snippet it's trying to convert the GThread Action into an int. I have zero insight as to what the minimum arguments are for it to process a simply void of code.

deepshit - December 5, 2018, 3:16 am

Send me your project solution and I'll try to fix it for you.

Or just the cudafy library.