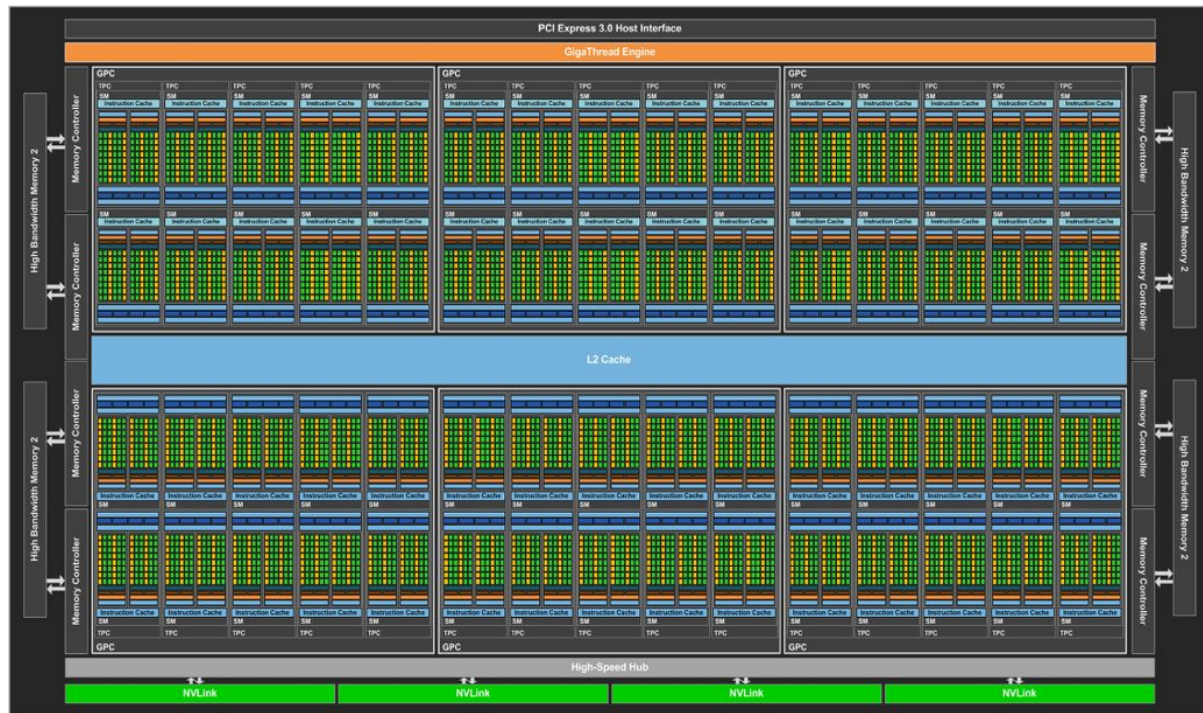


Programació paral·lela

Tema 2: CUDA, Device code

Estructura d'una GPU



Streaming Multiprocessor



Device code: Model de programació

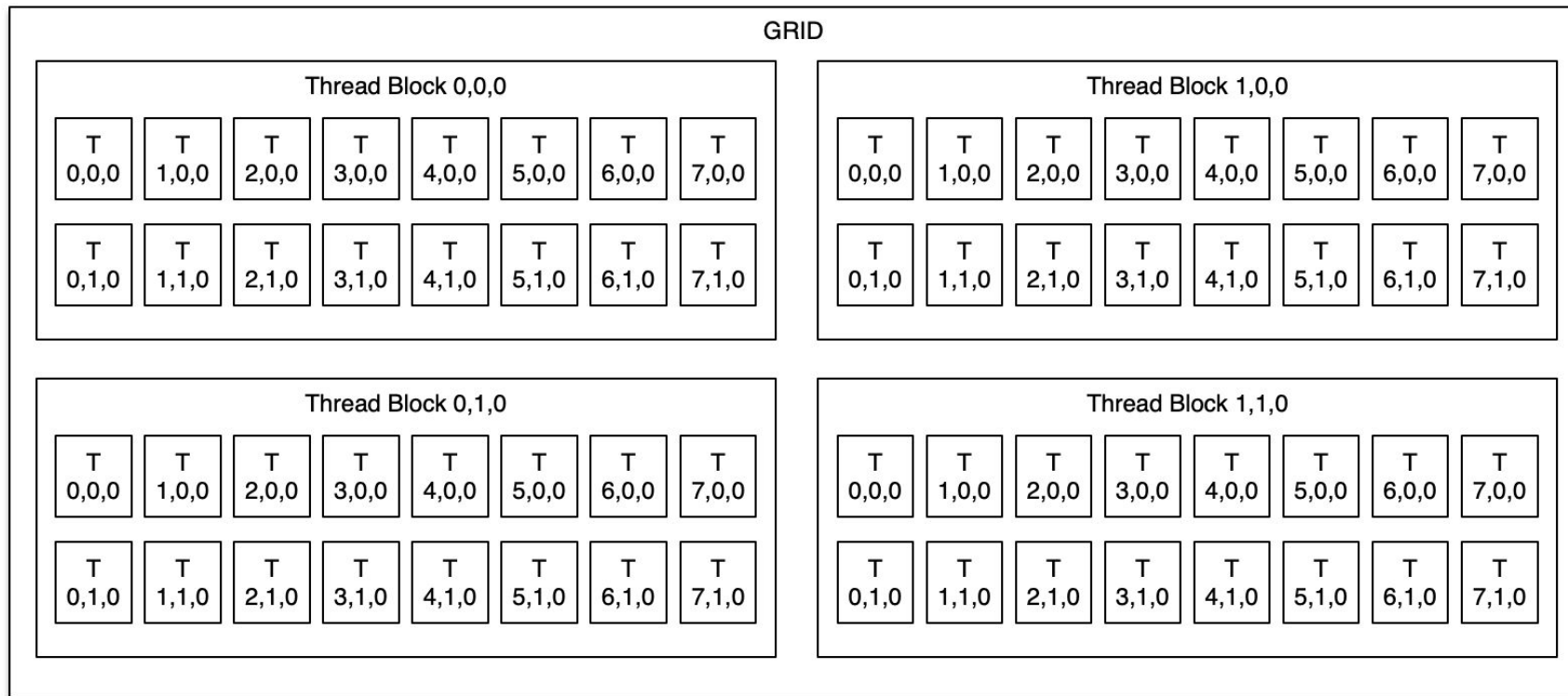
CPU:

```
for (int i=0; i<N; ++i) {  
    vector[i] = vector[i]*vector[i];  
}
```

GPU:

```
vector[threadIdx.x] = vector[threadIdx.x] * vector[threadIdx.x];
```

Device code: Model de programació



Device code: Model de programació

```
struct threadIdx { int x; int y; int z }; // Id de cada thread dins d'un block
```

```
struct blockIdx { int x; int y; int z }; // Id de cada block dins d'un grid
```

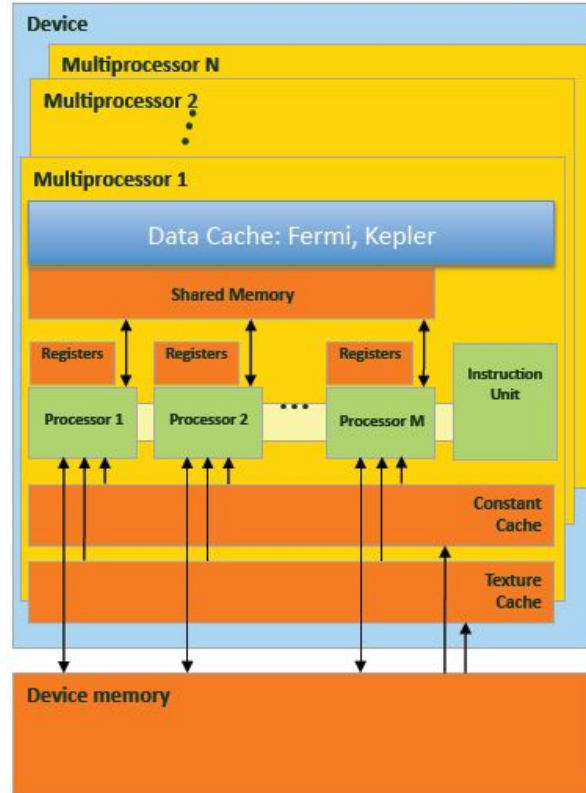
```
struct blockDim { int x; int y; int z }; // Número de threads dins d'un block
```

```
struct gridDim { int x; int y; int z }; // Número de blocks dins d'un Grid
```

Device code: Model de programació

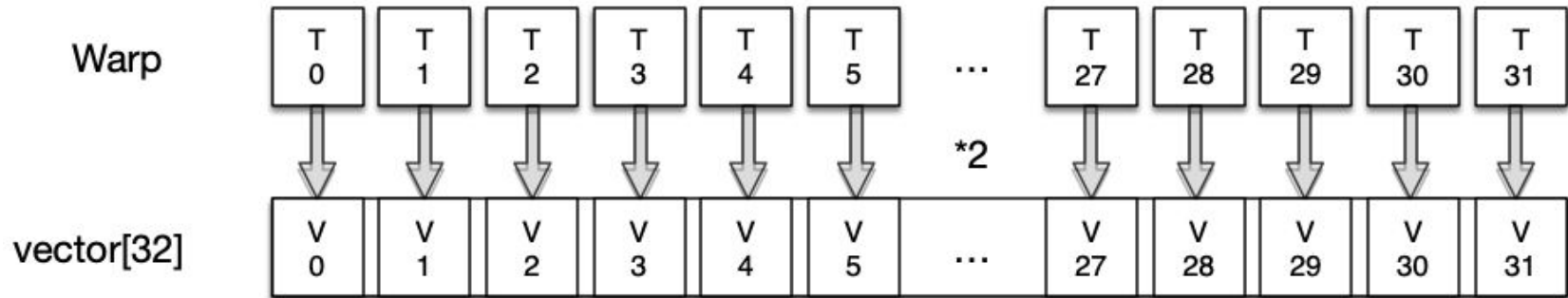
```
int vector[2048];  
int thread_position = threadIdx.x + blockIdx.x * blockDim.x;  
vector[thread_position] = vector[thread_position] * vector[thread_position];
```

Device code: jerarquia de memòries



Device code: model d'execució

`vector[threadIdx.x] *= 2;`



Device code: model d'execució

```
__syncthreads();
```

Coalescencia

