

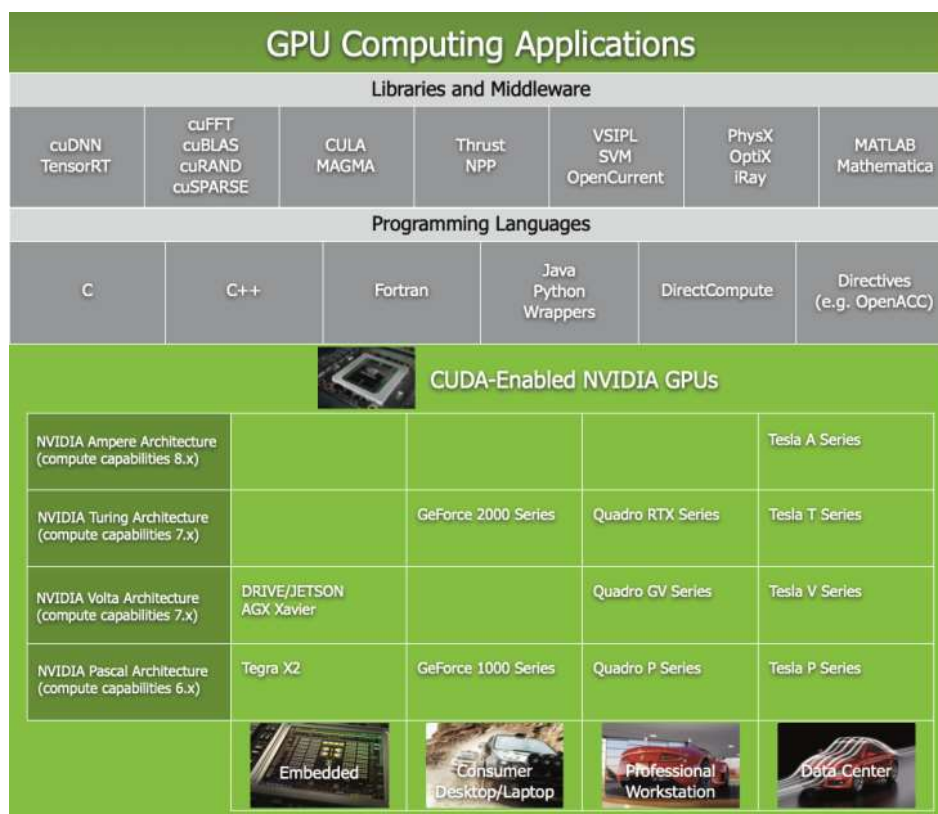
13 - дәріс. Графикалық процессордағы параллельді есептеу: CUDA. CUDA ағындары.

Жоспар:

- CUDA дегеніміз не?
- CUDA негізгі ұғымдары
- CUDA-да қарапайым бағдарлама мысалдары

CUDA дегеніміз не?

Графикалық процессорда параллельді есептеулерді орындау үшін NVIDIA компаниясының әзірлеушілері параллельді есептеулердің әмбебап архитектурасын – NVIDIA CUDA (Compute Unified Device Architecture) әзірледі. Ол ISA (Instruction Set Architecture) архитектуралық нұсқаулар жиынтығын және GPU-да параллельді есептеулерді енгізуді қамтиды. Ол көптеген бағдарламалау интерфейстерін қолдайды:



28 – сурет. GPU есептеуіш қолданбалары

CUDA жинағы:

- CUDA driver;
- CUDA Toolkit;
 - компилятор;
 - профиль жасаушы;
 - оңтайландырылған кітапханалар;
 - құжаттама.
- GPU Computing SDK;
- Негізгі операциялық жүйелерді қолдау.

CUDA негізгі ұғымдары

CUDA-ны бағдарламалық тұрғыдан қарастырмас бұрын, ағын, варп, блок және тор сияқты ұғымдарды ескеру қажет.

- Ағын (thread) – өңдеуге арналған негізгі деректер жиынтығы. Айта кету керек, CPU және GPU ағындары әртүрлі мәндерге ие. Бұрын айтылғандай, екі ағын арасында ауысу GPU-да ресурстарды қажет ететін операция емес.
- Варп (warp) – CUDA мультипроцессорларында SIMD әдісімен өңделетін деректердің ең аз көлемі. Басқаша айтқанда, бұл 32 ағыннан тұратын топ.
- Блок (block) – өзара байланысты ағындар тобы. Блоктардың максималды саны-65535.
- Тор (grid) – өңделуі керек блоктар жиынтығы. Желі үш өлшемді өлшемге ие болуы мүмкін.

GPU қолданатын бағдарлама 2 бөліктен тұрады:

- C / C++ коды (соның ішінде негізгі функция) әдеттегідей CPU-да орындалады.
- GPU-да ядролар және олардың ішінде шақырылатын функциялар арнайы функциялар орындалады.

Ядро ағындық функция болып табылады - көптеген ағындар ядро денесін параллель орындайды. Ядро CPU тарапынан шақырылады, оны орындайтын ағындардың саны көрсетіледі. GPU-дегі есептеу моделі көбінесе ағындық есептеу моделіне ұқсайды.

CUDA-да қарапайым бағдарлама мысалдары

GPU-ға екі массив қосатын бағдарламаның қарапайым мысалын қарастырайық:

```
#include <cuda_runtime.h>
#include <iostream>

__global__ void add(int *a, int *b, int *c) {
    int index = threadIdx.x;
    c[index] = a[index] + b[index];
}

int main() {
    const int arraySize = 5;
    int a[arraySize] = {1, 2, 3, 4, 5};
    int b[arraySize] = {10, 20, 30, 40, 50};
    int c[arraySize];

    int *d_a, *d_b, *d_c;
    size_t size = arraySize * sizeof(int);

    // Құрылғыда сақтау орнын бөлу
    cudaMalloc((void**)&d_a, size);
    cudaMalloc((void**)&d_b, size);
    cudaMalloc((void**)&d_c, size);
```

```

// Деректерді хосттан құрылғыға көшіру
cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);

// Ядроны іске қосу
add<<<1, arraySize>>>(d_a, d_b, d_c);

// Нәтижелерді құрылғыдан хостқа көшіру
cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);

// Нәтижелерді шығару
std::cout << "Result: ";
for (int i = 0; i < arraySize; i++) {
    std::cout << c[i] << " ";
}
std::cout << std::endl;

// Жадыны босату
cudaFree(d_a);
cudaFree(d_b);
cudaFree(d_c);

return 0;
}

```

CUDA ағындарын пайдалану мысалы. CUDA ағындарын қолдана отырып, массивтің әр элементін 2-ге көбейтетін бағдарламаның мысалын қарастырайық:

```

#include <cuda_runtime.h>
#include <iostream>

__global__ void multiply(int *a, int factor) {
    int index = threadIdx.x;
    a[index] *= factor;
}

int main() {
    const int arraySize = 5;
    int a[arraySize] = {1, 2, 3, 4, 5};

    int *d_a;
    size_t size = arraySize * sizeof(int);

    // Құрылғыдан орын бөлу

```

```

cudaMalloc((void**)&d_a, size);

// Деректерді хосттан құрылғыға көшіру
cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);

// Ядроны іске қосу
multiply<<<1, arraySize>>>(d_a, 2);

// Нәтижелерді құрылғыдан хостқа көшіру
cudaMemcpy(a, d_a, size, cudaMemcpyDeviceToHost);

// Нәтижелерді шығару
std::cout << "Result: ";
for (int i = 0; i < arraySize; i++) {
    std::cout << a[i] << " ";
}
std::cout << std::endl;

// Жадыны босату
cudaFree(d_a);

return 0;
}

```

Тапсырма:

1. Дәрісте келтірілген CUDA бағдарламаларын өз компьютеріңізде орындаңыз.

Бақылау сұрақтары:

1. CUDA дегеніміз не және ол қандай негізгі компоненттерді қамтиды?
2. CUDA-да жадты бөлу және босату процесін түсіндіріңіз.
3. CUDA-да ағындар мен блоктар қалай ұйымдастырылған?