Вычисления на GPU

CPU

- Число ядер:

  единицы, 8-16...
- Дополнительно:

  SSE, MMX, AVX
- Производительность

  ~100 GFlops

MIC

- Число ядер:

  десятки, 32-50...

- Дополнительно:

  OpenMP, OpenCL, Cilk

- Производительность

  ~ TFlop

MIC example

```
#define NSET 1000000

int main(int argc, const char** argv)
{
  long int i;
  float Pi, num_inside = 0.0f;

#pragma offload target (MIC)
#pragma omp parallel for reduction(+:num_inside)
  for(i = 0; i < NSET; i++) {
    float x = float(rand()) / float(RAND_MAX + 1);
    float y = float(rand()) / float(RAND_MAX + 1);
    float distance_from_zero = sqrt(x * x + y * y);

    if(distance_from_zero <= 1.0f)
      num_inside += 1.0f;
  }

  Pi = 4.0f * (num_inside / NSET);
  printf("Value of Pi = %f \n", Pi);
}
```

GPU

- Число ядер:

  сотни, 240...

- Дополнительно:

- Производительность

  ~TFlop

Библиотеки

Intel:

- *IPP (Integrated Performance Primitives)*
- *MKL (Math Kernel Library)*

NVIDIA:

- *NPP (NVIDIA Performance Primitives)*
- *cuBLAS (CUDA Basic Linear Algebra Subroutines)*

AMD:

- *APPML (Accelerated Parallel Processing Math Lib.)*
- *ACML (AMD Core Math Library)*

Brook+

- Поддержка:

  AMD Stream, сейчас Open Source проект

- Что это:

  C компилятор с программированием в терминах ядер и потоков

- Пример:

```
kernel void sumaa(float a<>, float b<>, out float c<>){
    c = a + b;
}
```

CUDA

- Поддержка:

  NVIDIA, используется множеством проектов (Folding@Home с 2008 г...)

- Что это: *API для C/C++, FORTRAN*
- Пример:

```
__global__ void VecAdd(const float* A, const float* B,
float* C, int N) {
    int i = blockDim.x * blockIdx.x + threadIdx.x;
    if (i < N)
        C[i] = A[i] + B[i];
}
```

OpenCL

- Поддержка:

  AMD, NVIDIA, Intel

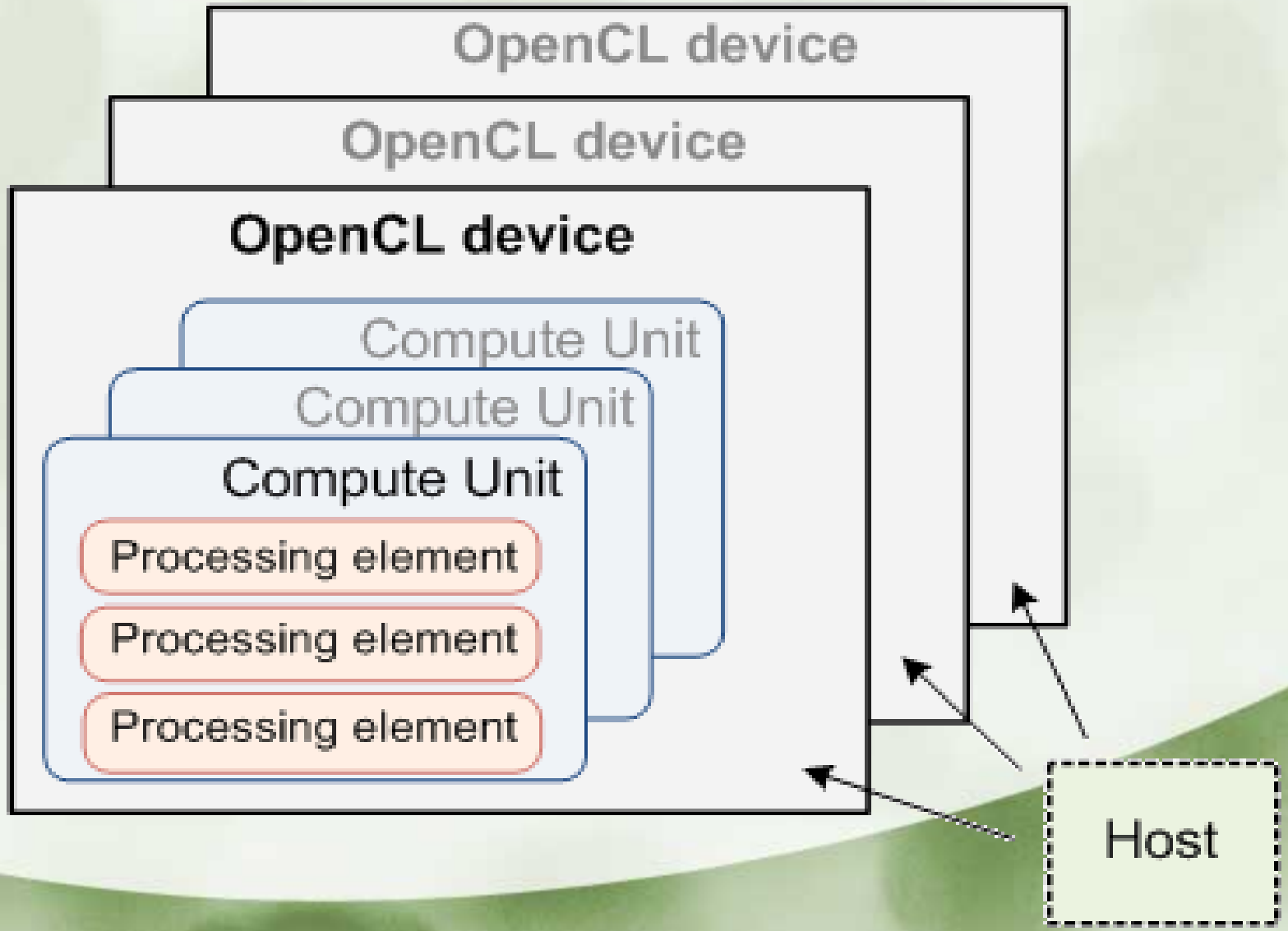
- Что это:

  Стандарт вычислений для гетерогенных систем

- Пример:

```
__kernel void VectorAdd(__global const float* a,
__global const float* b, __global float* c, int
iNumElements) {
   int iGID = get_global_id(0);
   c[iGID] = a[iGID] + b[iGID];
}
```

Модель вычислений OpenCL



Гибридные вычисления

- OpenCL
- #pragma omp target device (fpga)
- Интегрированные графические ядра:
  - AMD Accelerated Processing Unit
  - Intel Sandy Bridge