Effective Use of Accelerators

Blue Waters Petascale Project

Wen-Mei Hwu

May 11, 2015

The IMPACT Research Group

ECE ILLINOIS

ILLINOIS

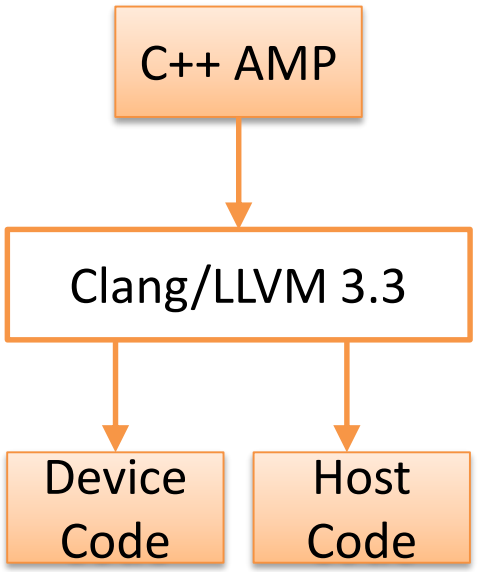
C++AMP Overview

```
void MultiplyWithAMP(int* aMatrix, int* bMatrix, int *productMatrix) {
    array_view<int, 2> a(3, 2, aMatrix);
    array_view<int, 2> b(2, 3, bMatrix);
    array_view<int, 2> product(3, 3, productMatrix);
    parallel_for_each(
        product.extent,
        [=](index<2> idx) restrict(amp)
            int row = idx[0];
            int col = idx[1];
            for (int inner = 0; inner < a.get_extent()[1]; inner++)
                product[idx] += a(row, inner) * b(inner, col);
    );
    product.synchronize();
}
```

GPU data modeled as data container

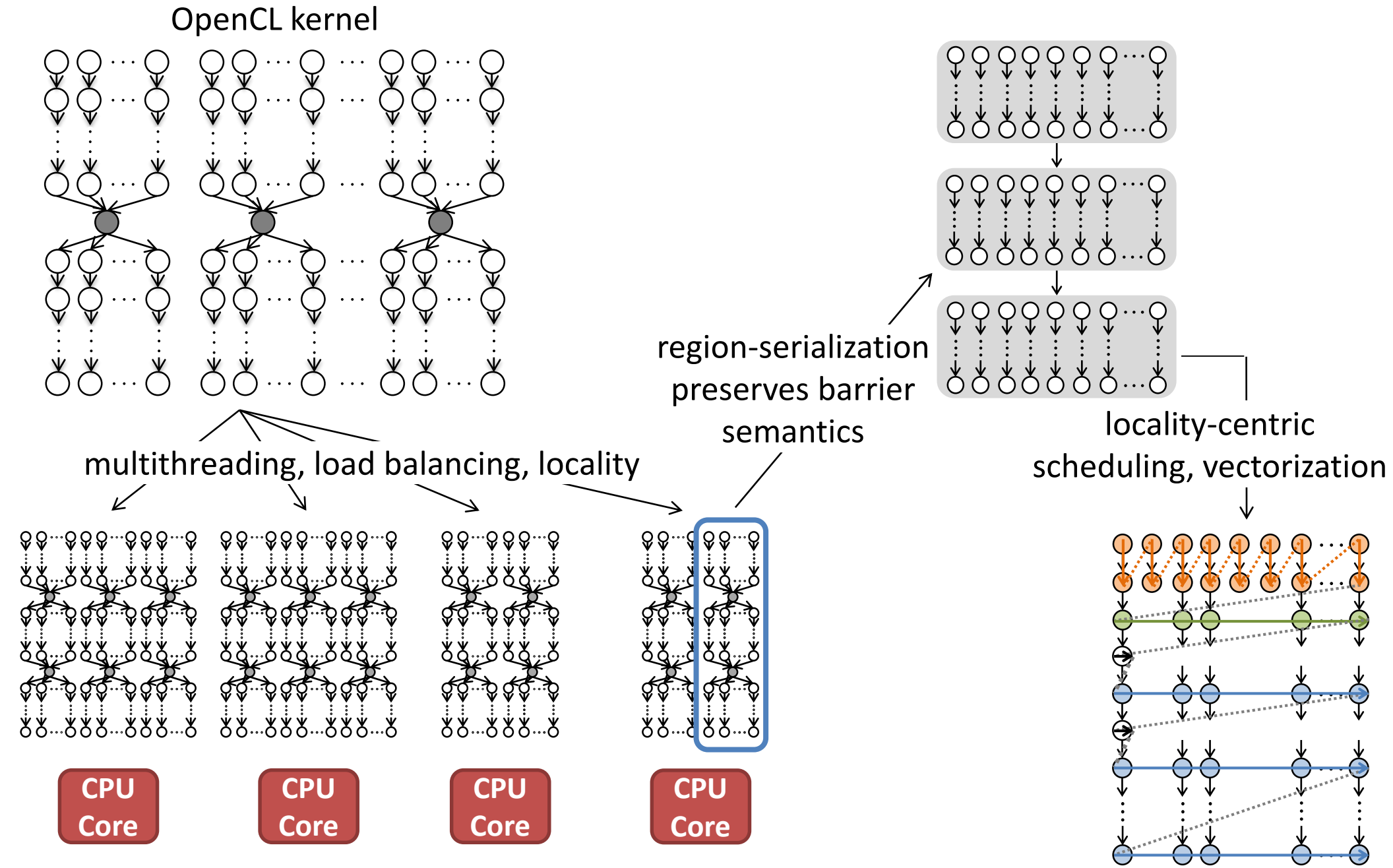
Execution interface; marking an implicitly parallel region for GPU execution

Kernels modeled as lambdas; arguments are implicitly modeled as captured variables

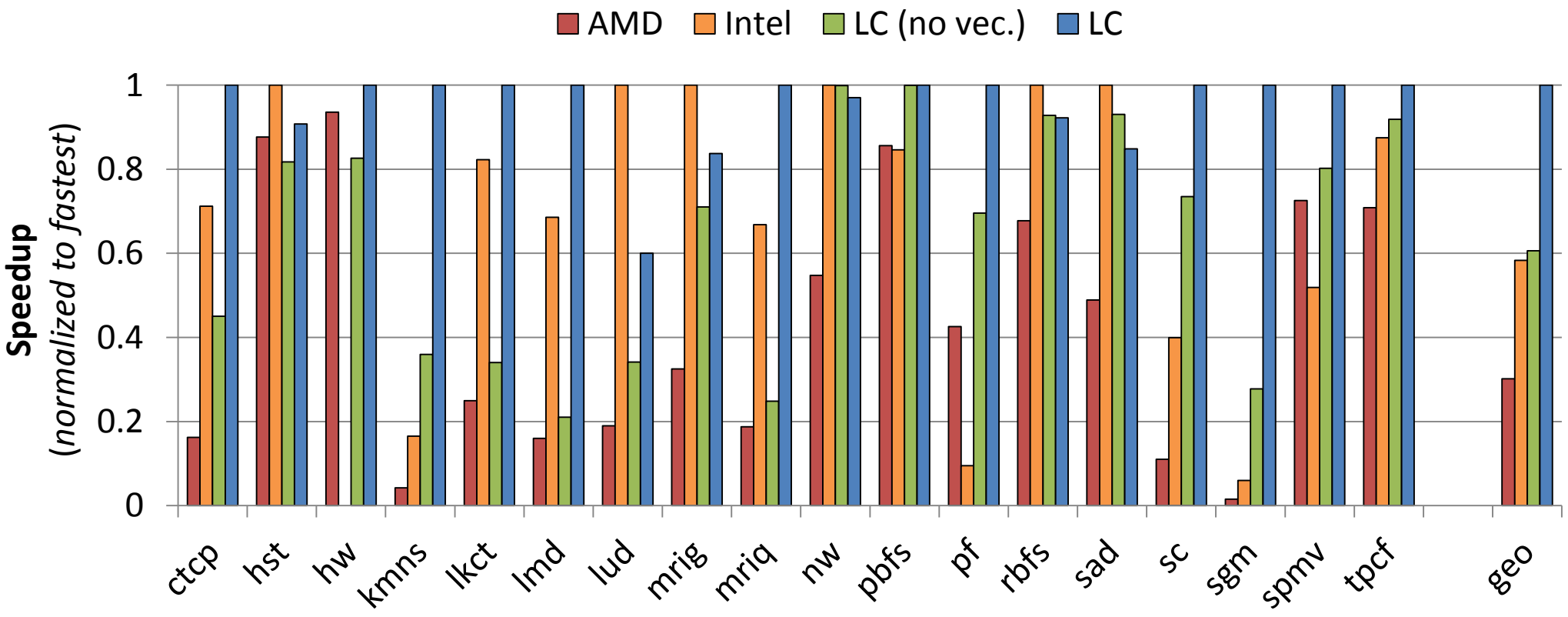


- Device Path
  - generate OpenCL C code by CBackend
  - emit kernel function
- Host Path
  - preparation to launch the code

MxPA Overview



MxPA (LC) Performance



LC (with vec.) outperforms AMD (without vec.) and Intel (with vec.) by 3.32x and 1.71x

LC (without vec.) is faster than Intel (with vec.) by 1.04x

Other Compilers and Libraries

OpenACC

- Compiler directives to identify which areas of code to accelerate, without requiring modification to the underlying code itself.

- Allow the compiler to do the detailed work of mapping the computation onto the accelerator.

NVIDIA Thrust C++:

- GPU computing through the standard C++ template interface.

- Provides a flexible, high-level interface for GPU programming that greatly enhances developer productivity.

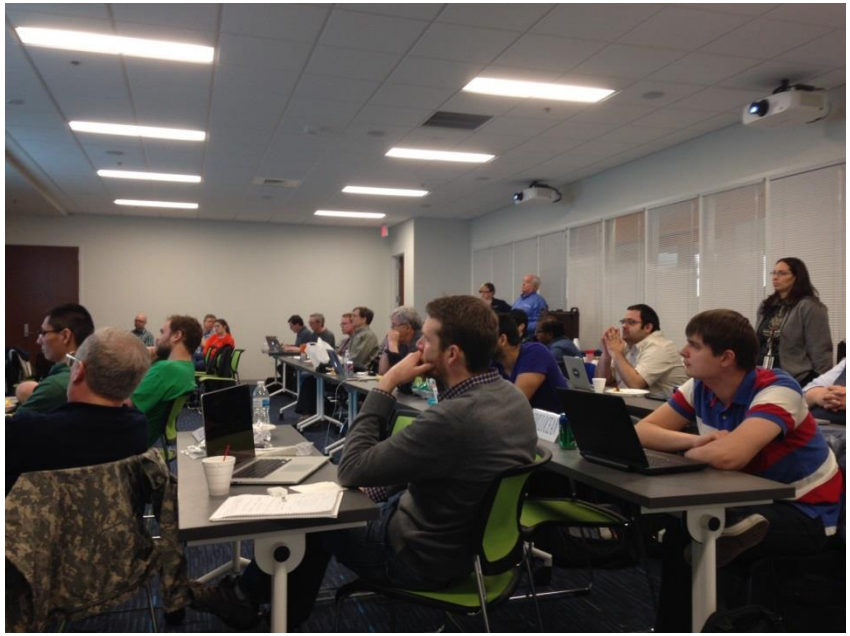
Working with IMPACT

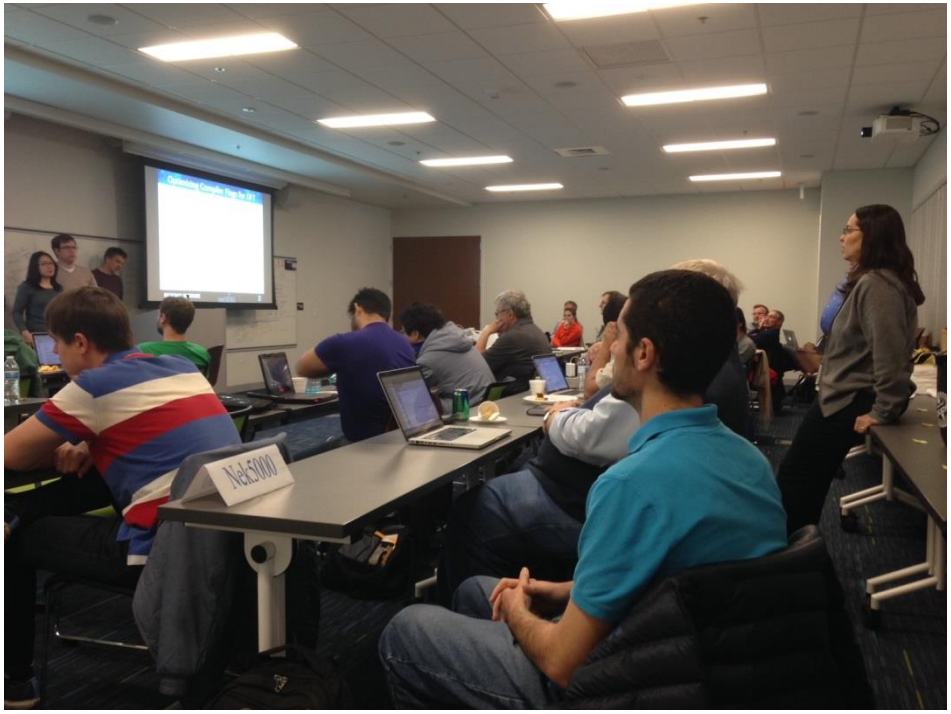
Experience in Education and Collaboration

- PUMPS summer schools.
- Porting real-world DOE applications.

Deep Knowledge of Devices, Programming, and Compilers

- Expertise with heterogeneous devices.
- Close-to-metal programming for performance-critical libraries.





Objective of Collaboration

More science in less time.

C++ AMP, MxPA, Thrust, Cray, PGI

- Significant reduction in heterogeneous cluster programming complexity.

Improving community tools and libraries.

Questions?

Thanks for your time