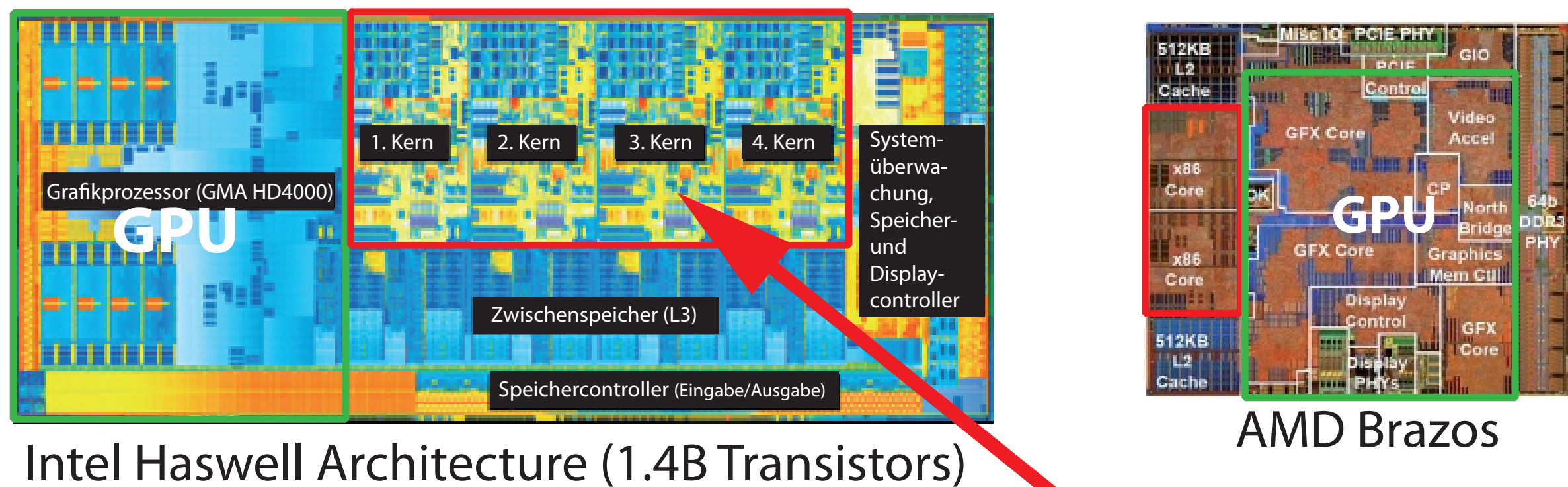


# AnyDSL: A COMPILER-FRAMEWORK FOR DOMAIN-SPECIFIC LIBRARIES (DSLs)

Richard Membarth, Arsène Pérard-Gayot, Martin Weier, Philipp Slusallek  
Roland Leißa, Klaas Boesche, Sebastian Hack

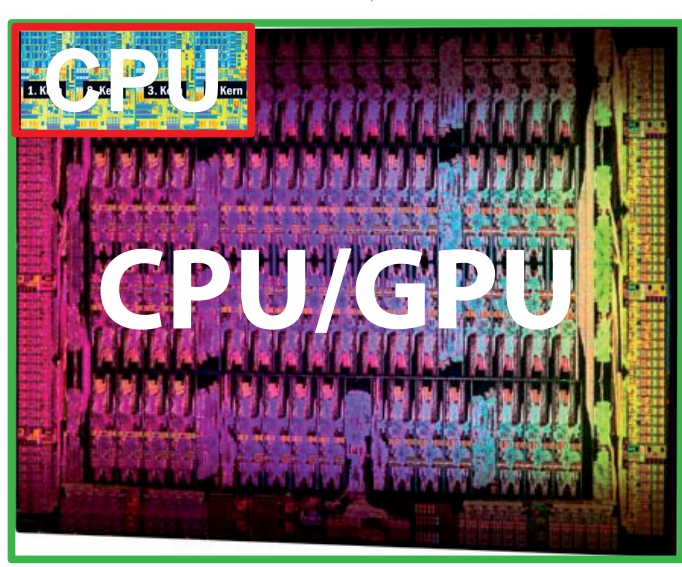
## Motivation

- Many-Core HW is everywhere
- But cannot be programmed well

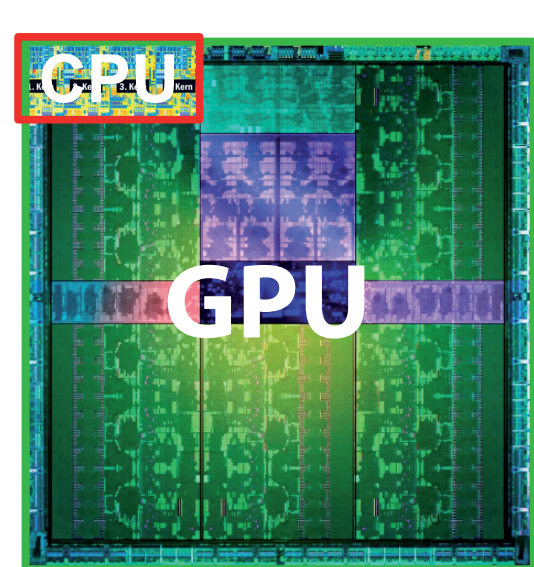


Intel Haswell Architecture (1.4B Transistors)

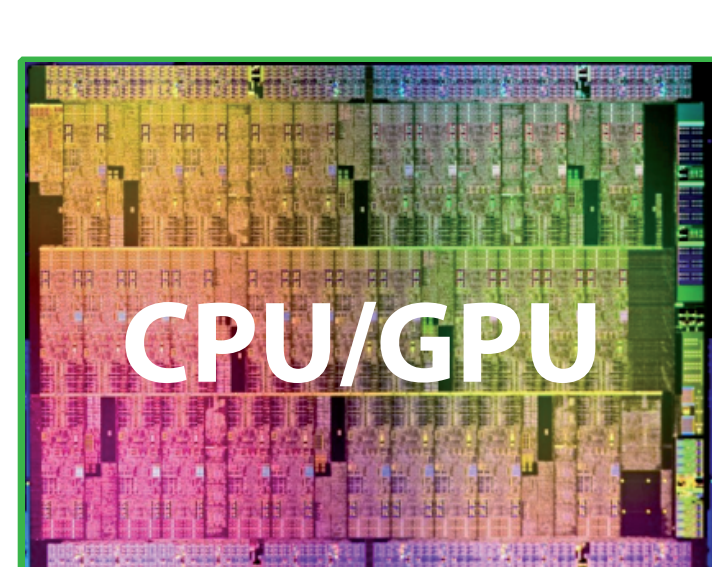
AMD Brazos



Intel KnightsFerry  
(~5B Transistors)



Nvidia Kepler  
(~7B Transistors)



Intel Knights Landing

Traditional Programs run  
only on a single core

## Embedding of DSLs in Impala

- Separation of concerns through code refinement
  - Higher-order functions
  - Partial evaluation
  - Triggered code generation

### Application Developer

```
fn main() {
  let img = load("dragon.png");
  let blurred = gaussian_blur(img);
}
```

### DSL Developer

```
fn gaussian_blur(field: Field) -> Field {
  let stencil: Stencil = { /* ... */ };
  let mut out: Field = { /* ... */ };

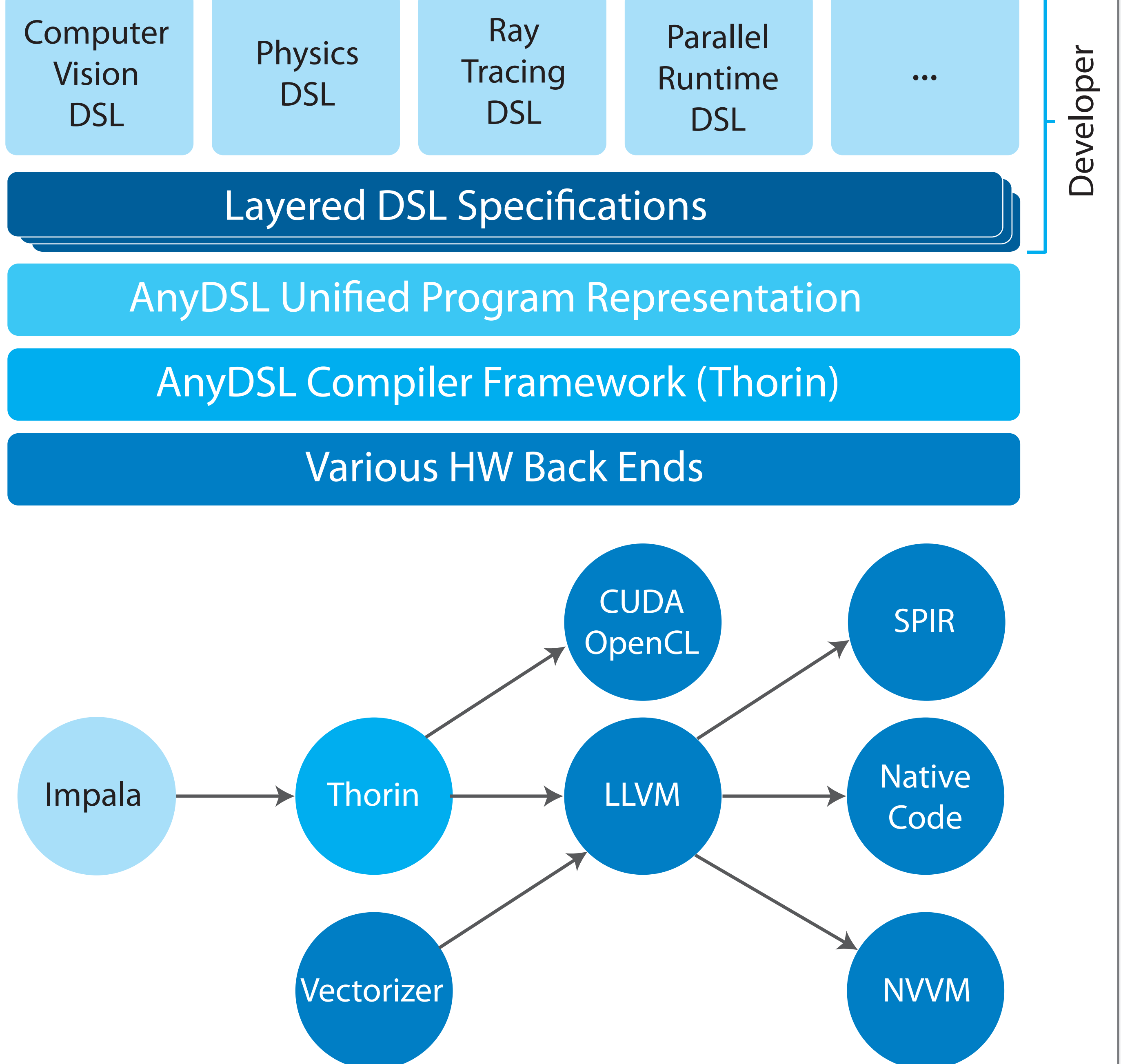
  for x, y in @iterate(out) {
    out.data(x, y) = apply_stencil(x, y, field, stencil);
  }
  out
}
```

### Machine Expert

```
fn iterate(field: Field, body: fn(int, int) -> () -> () {
  let grid = (field.cols, field.rows, 1);
  let block = (128, 1, 1);

  with nvvm(grid, block) {
    let x = nvvm_tid_x() + nvvm_ntid_x() * nvvm_ctaid_x();
    let y = nvvm_tid_y() + nvvm_ntid_y() * nvvm_ctaid_y();
    body(x, y);
  }
}
```

## AnyDSL Architecture



## Stincilla

- A DSL for stencil codes
- Example: Gaussian blur filter
  - Reference: OpenCV 3.0
  - Intel CPU: 40% faster
  - Intel GPU: 25% faster
  - AMD GPU: 50% faster
  - NVIDIA GPU: 45% faster
  - Up to 10x shorter code



## RaTrace

- A DSL for ray traversal
- 11% faster than Embree (on average, Core i7-4790)
- 17% faster than Aila et al. (on average, GTX 970)
- 1/10th of coding time (according to Halstead measures)

