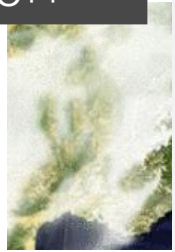
Polly-ACC: Transparent Compilation to Heterogeneous Hardware

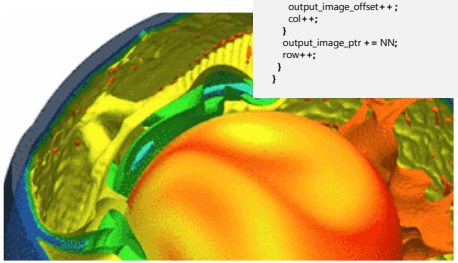
Tobias Grosser, Torsten Hoefler

Sequential Software

Fortran
C/C++

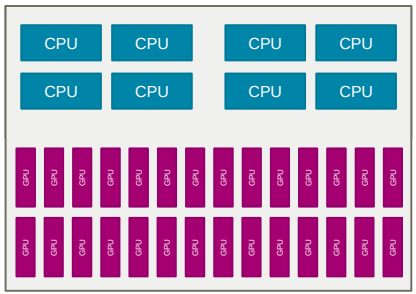


```
row = 0;
output_image_ptr = output_image;
output_image_ptr += (NN * dead_rows);
for (r = 0; r < NN - KK + 1; r++) {
  output_image_offset = output_image_ptr;
  output_image_offset += dead_cols;
  col = 0;
  for (c = 0; c < NN - KK + 1; c++) {
    input_image_ptr = input_image;
    input_image_ptr += (NN * row);
    kernel_ptr = kernel;
S0: *output_image_offset = 0;
    for (i = 0; i < KK; i++) {
      input_image_offset = input_image_ptr;
      input_image_offset += col;
      kernel_offset = kernel_ptr;
      for (j = 0; j < KK; j++) {
S1:     temp1 = *input_image_offset++;
S1:     temp2 = *kernel_offset++;
S1:     *output_image_offset += temp1 * temp2;
      }
      kernel_ptr += KK;
      input_image_ptr += NN;
    }
S2: *output_image_offset = ((*output_image_offset)/
normal_factor);
    output_image_offset++;
    col++;
  }
  output_image_ptr += NN;
  row++;
 }
}
```

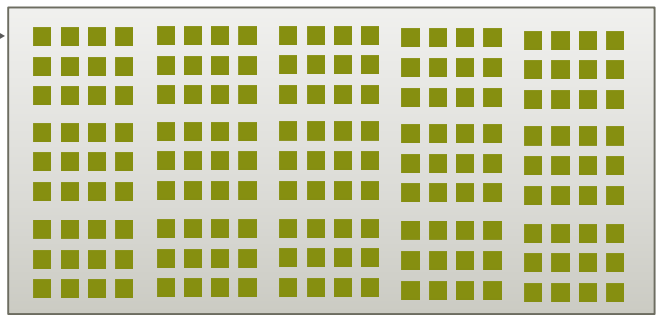


Parallel Hardware

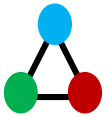
Multi-Core CPU

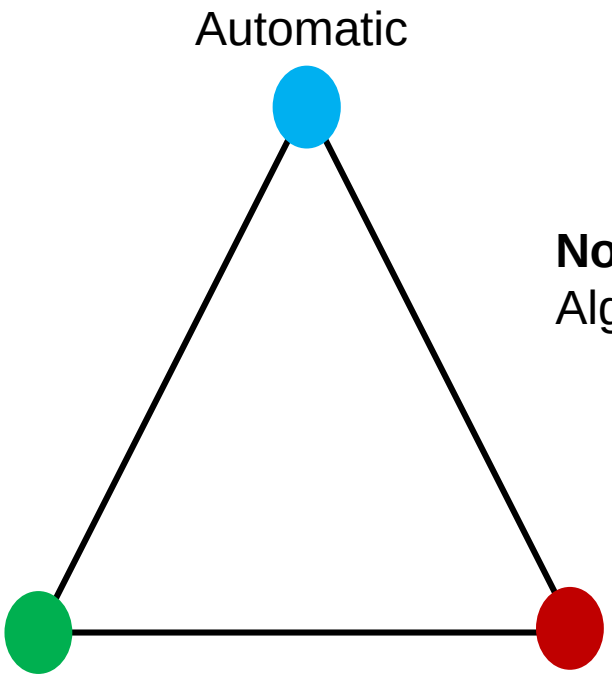


Accelerator



Design Goals





Non-Goal:
Algorithmic Changes

“Regression Free”

High Performance

Tool: Polyhedral Modeling

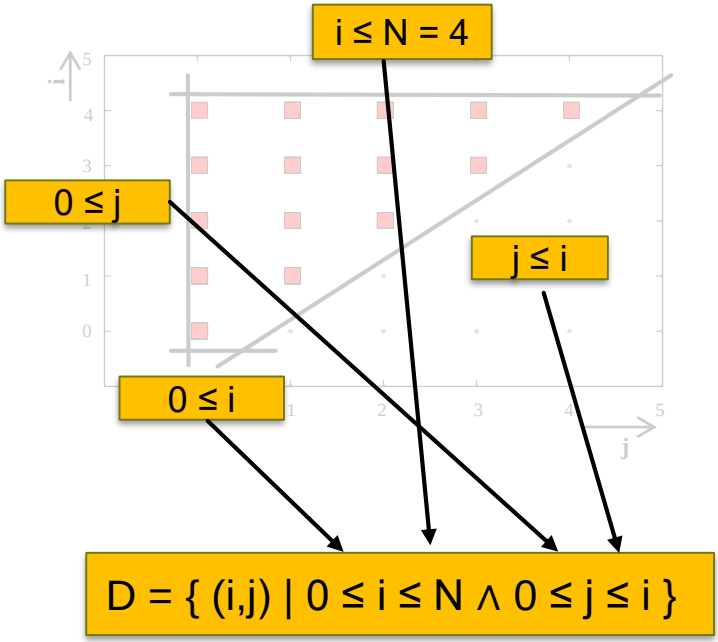
Program Code

```
for (i = 0; i <= N; i++)
  for (j = 0; j <= i; j++)
    S(i,j);
```

N = 4

(i, j) = (4,4)

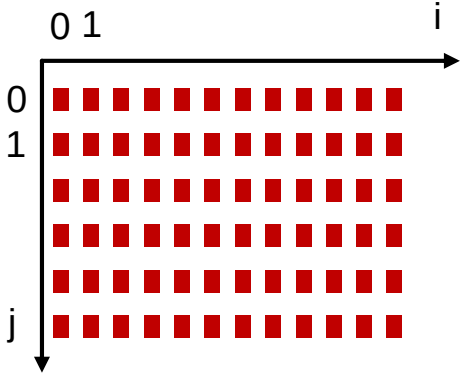
Iteration Space



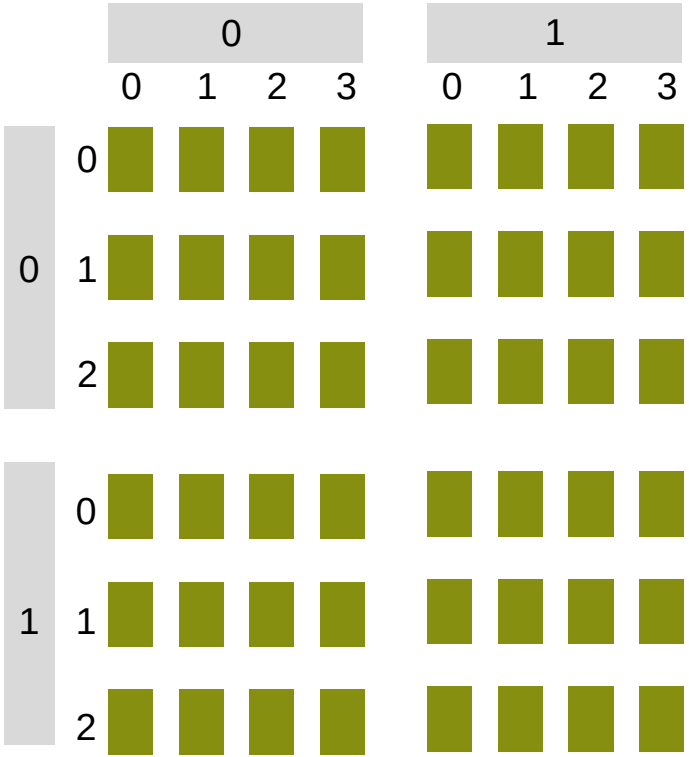
Polly -- Performing Polyhedral Optimizations on a Low-Level Intermediate Representation
Tobias Grosser et al,
Parallel Processing Letter, 2012

Mapping Computation to Device

Iteration Space



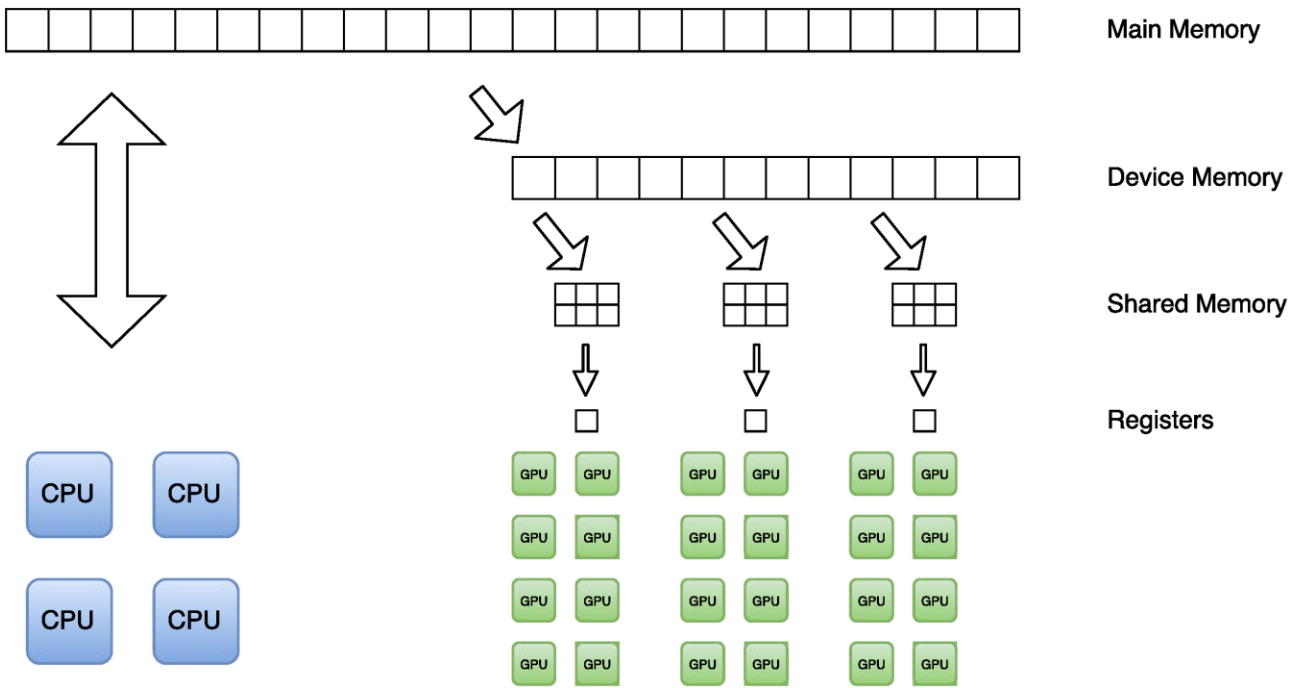
Device Blocks & Threads



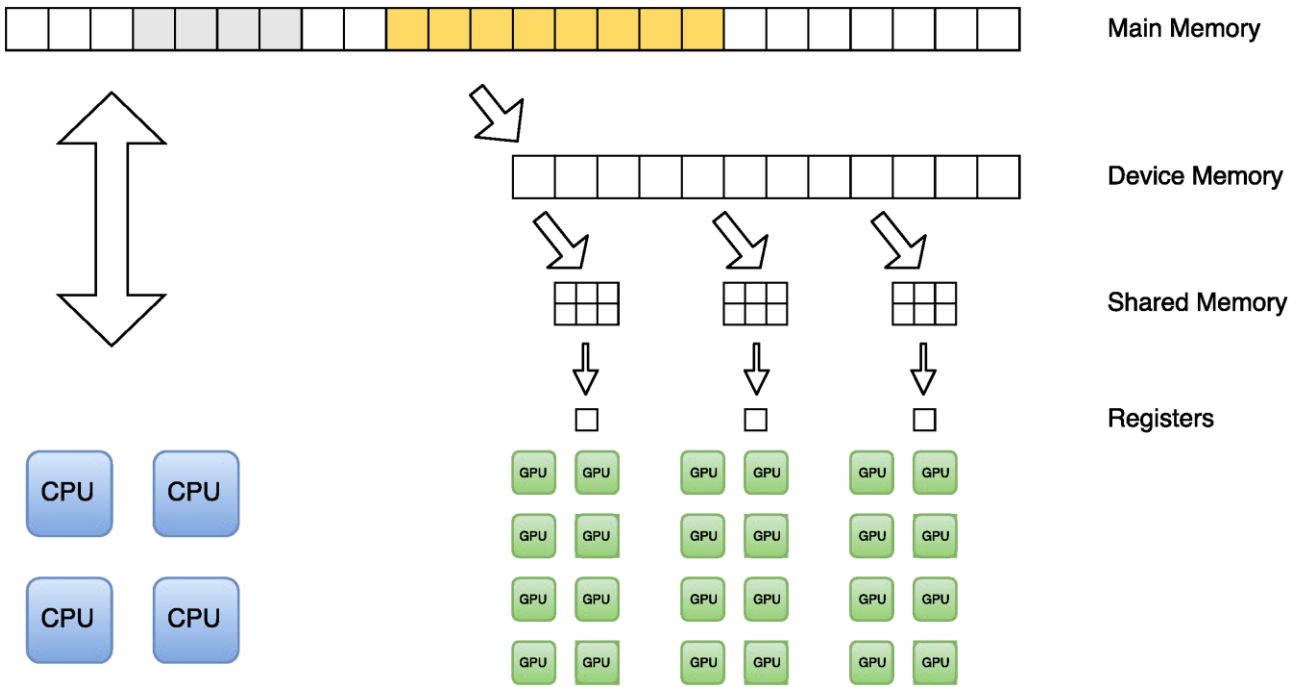
$$BID = \{(i,j) \rightarrow \left(\left\lfloor \frac{i}{4} \right\rfloor \% 2, \left\lfloor \frac{j}{3} \right\rfloor \% 2\right)\}$$

$$TID = \{(i,j) \rightarrow (i \% 4, j \% 3)\}$$

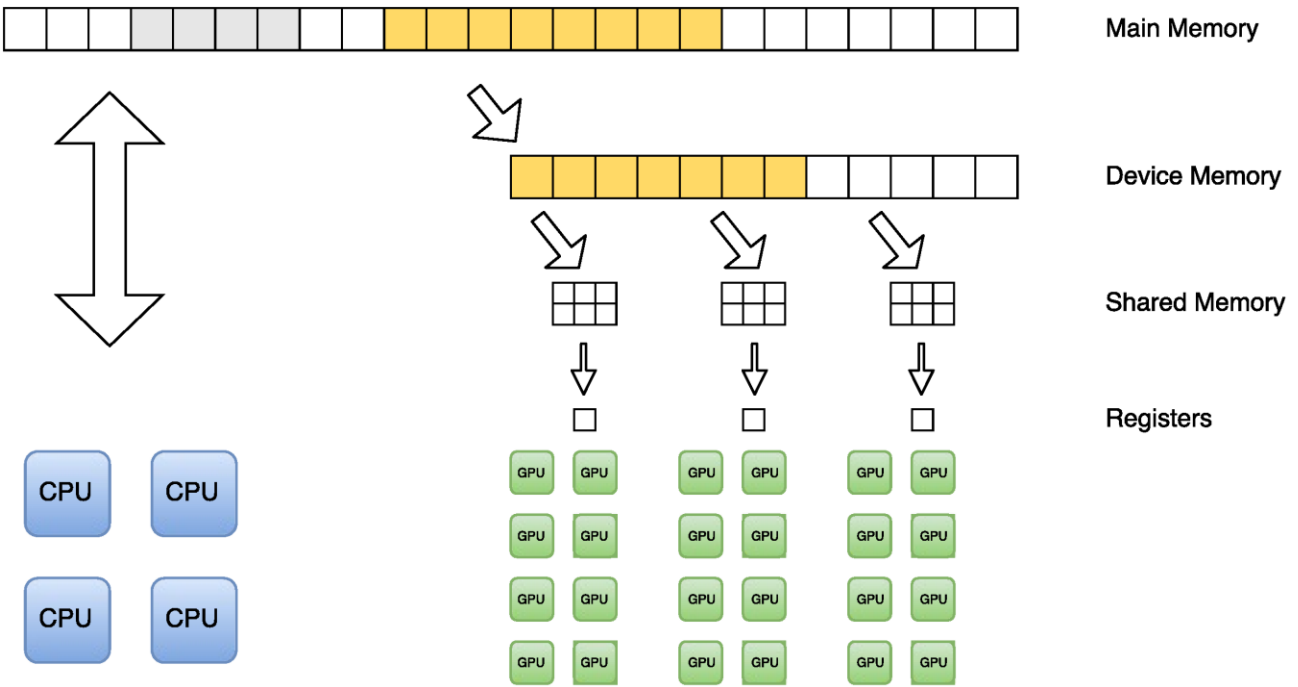
Memory Hierarchy of a Heterogeneous System



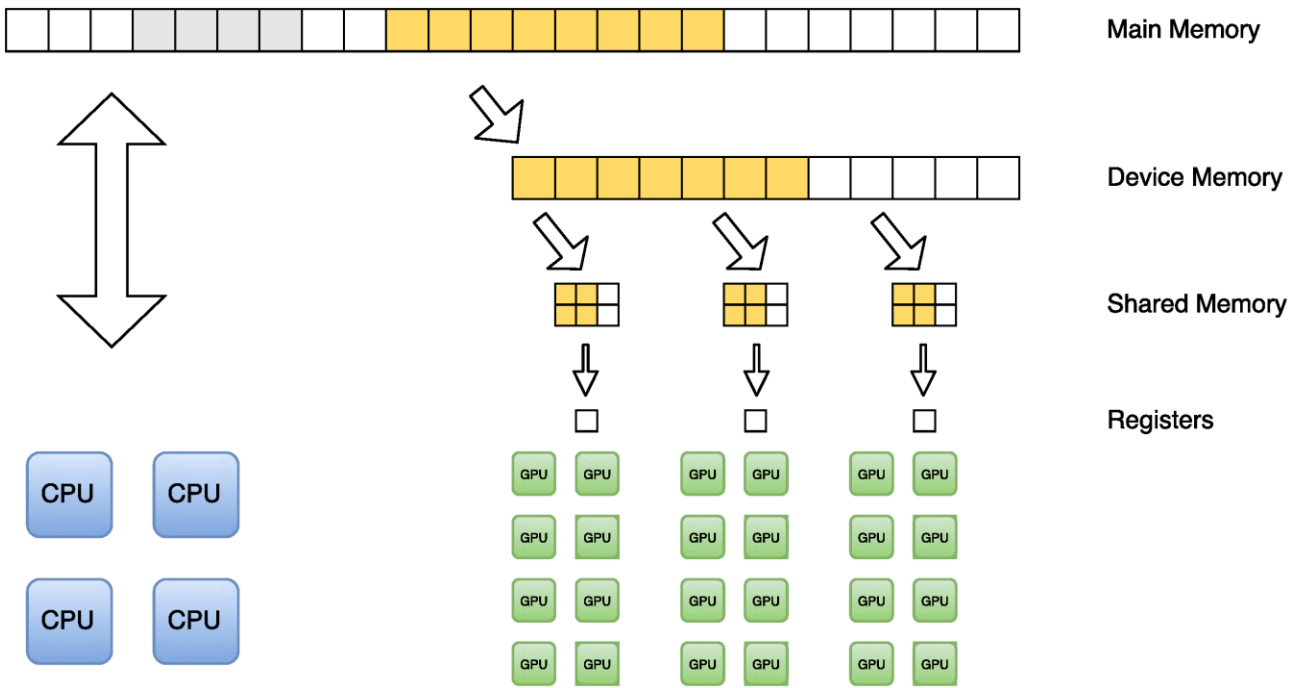
Host-device date transfers



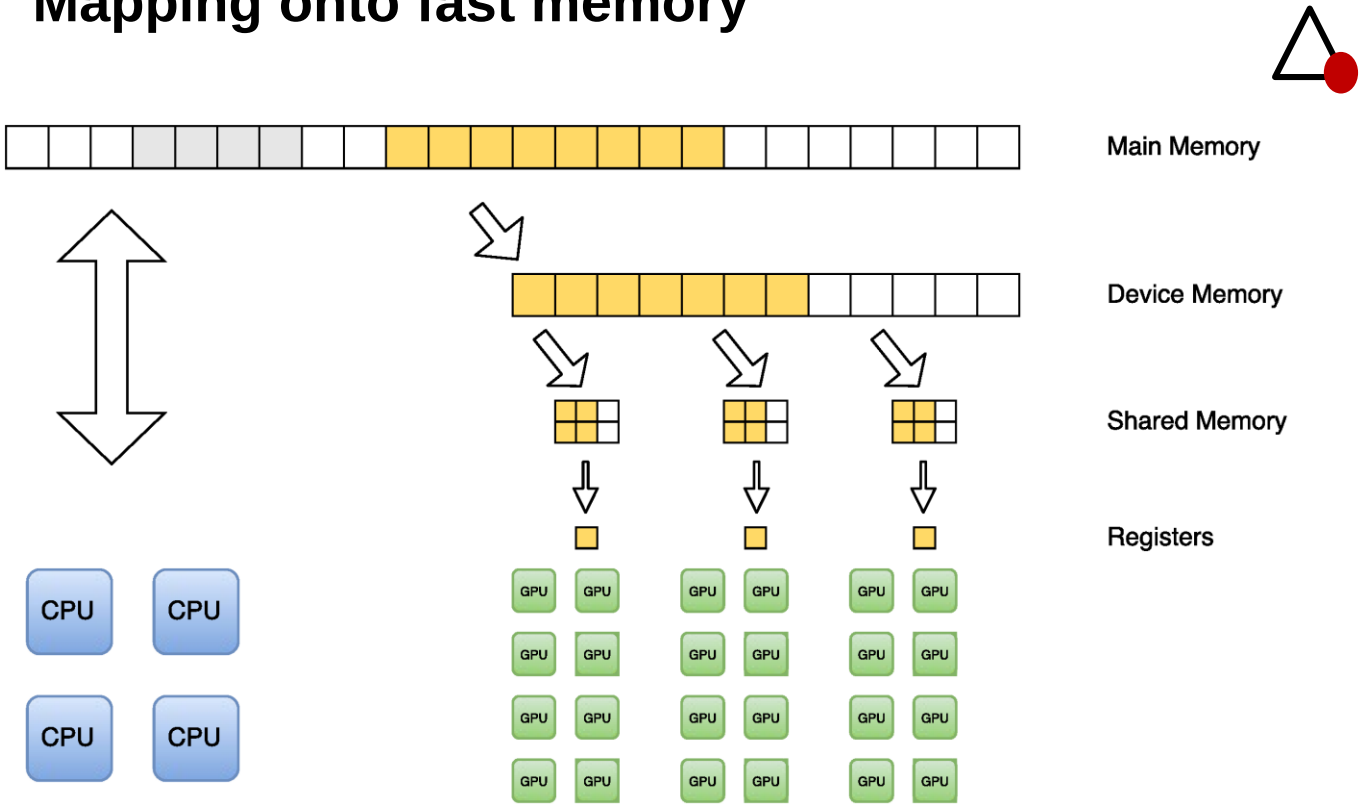
Host-device date transfers



Mapping onto fast memory

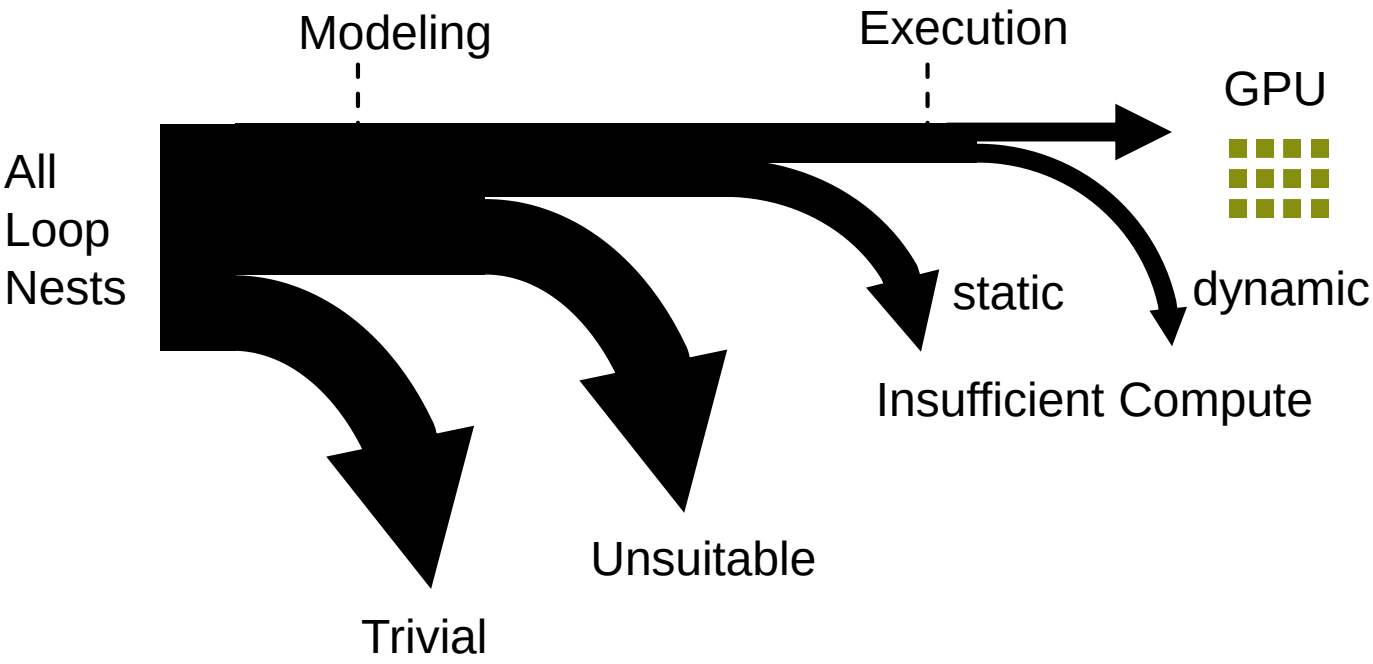


Mapping onto fast memory



Polyhedral parallel code generation for CUDA, Verdoolaege, Sven et. al, ACM Transactions on Architecture and Code Optimization, 2013

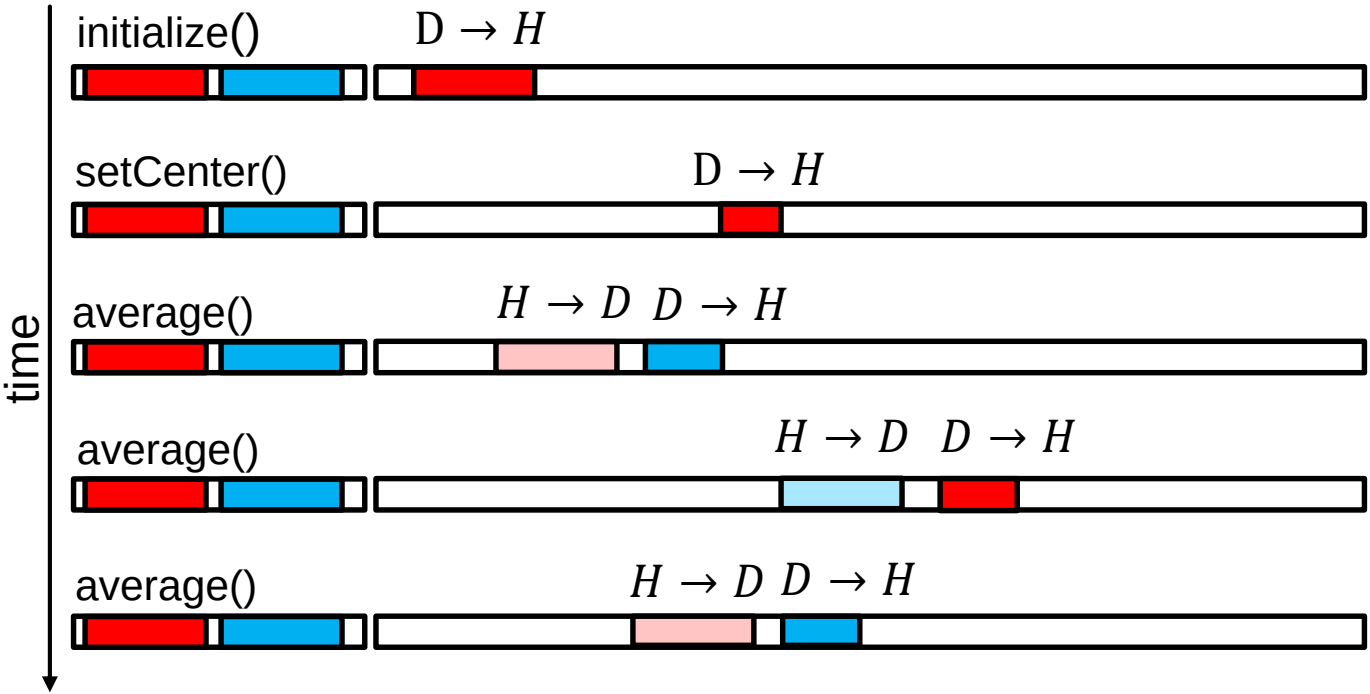
Profitability Heuristic



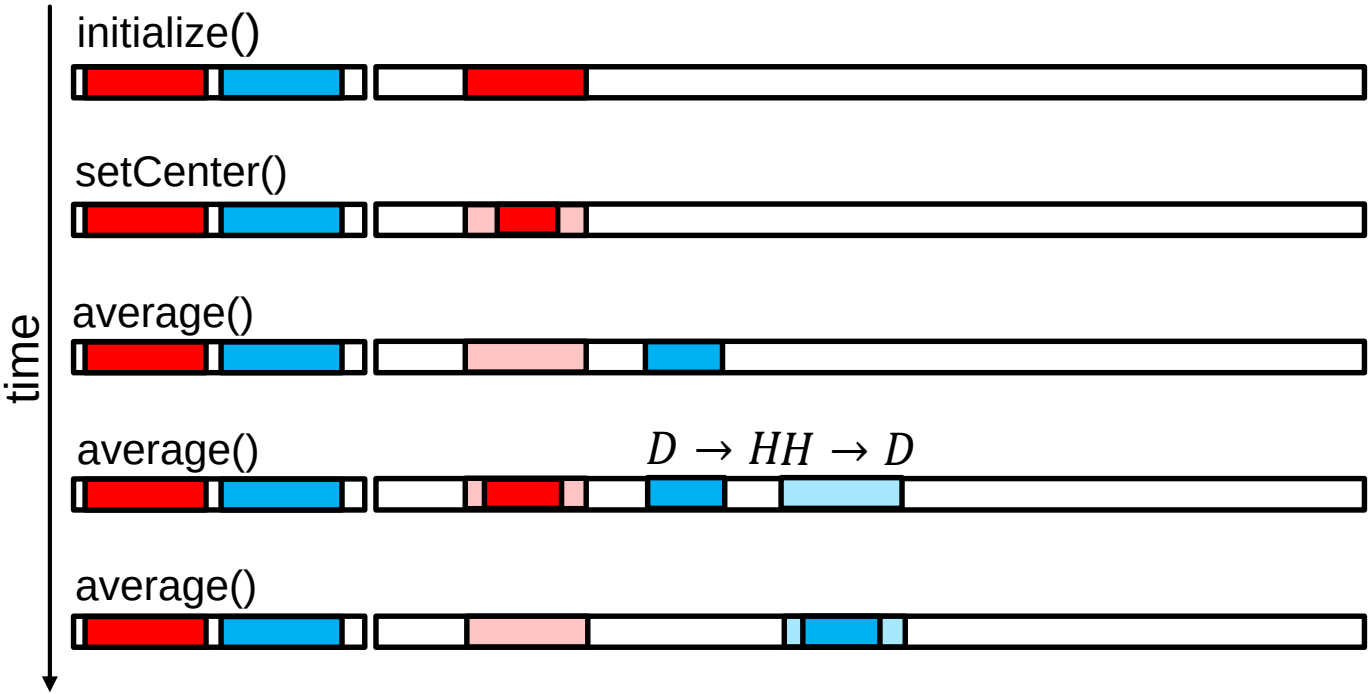
From kernels to program – data transfers

```
void heat(int n, float A[n], float hot, float cold) {

    float B[n] = {0};

    initialize(n, A, cold);
    setCenter(n, A, hot, n/4);

    for (int t = 0; t < T; t++) {
        average(n, A, B);
        average(n, B, A);
        printf("Iteration %d done", t);
    }
}
```

Data Transfer – Per Kernel



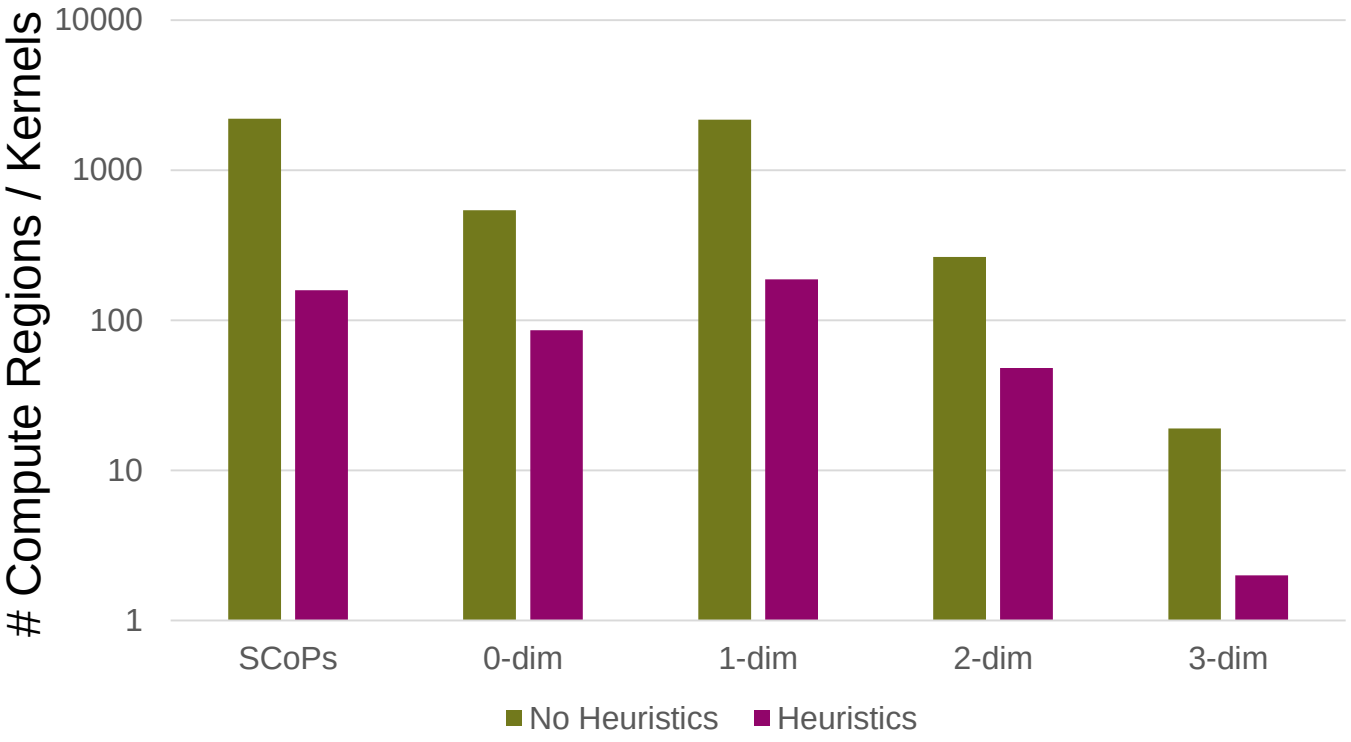
Data Transfer – Inter Kernel Caching



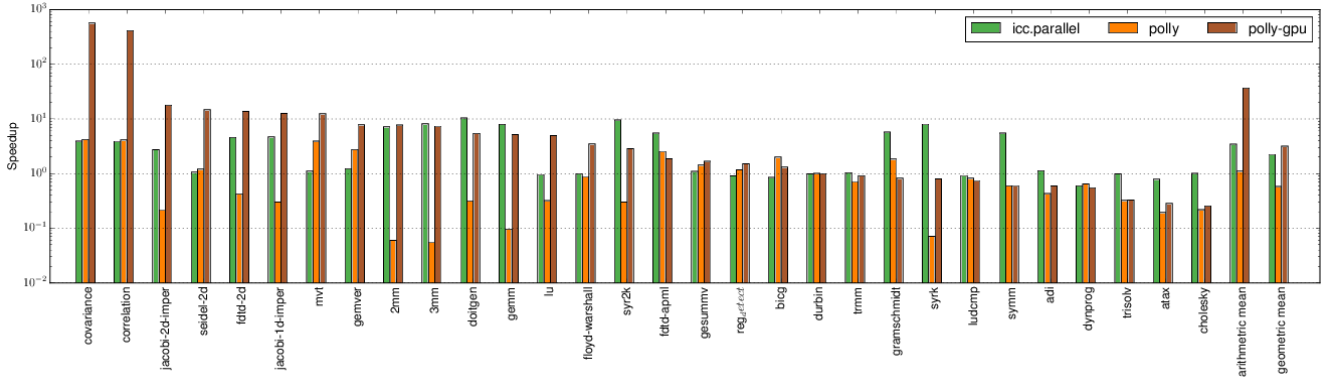
Evaluation

Workstation: 10 core SandyBridge NVIDIA Titan Black (Kepler)
Mobile: 4 core Haswell NVIDIA GT730M (Kepler)

LLVM Nightly Test Suite

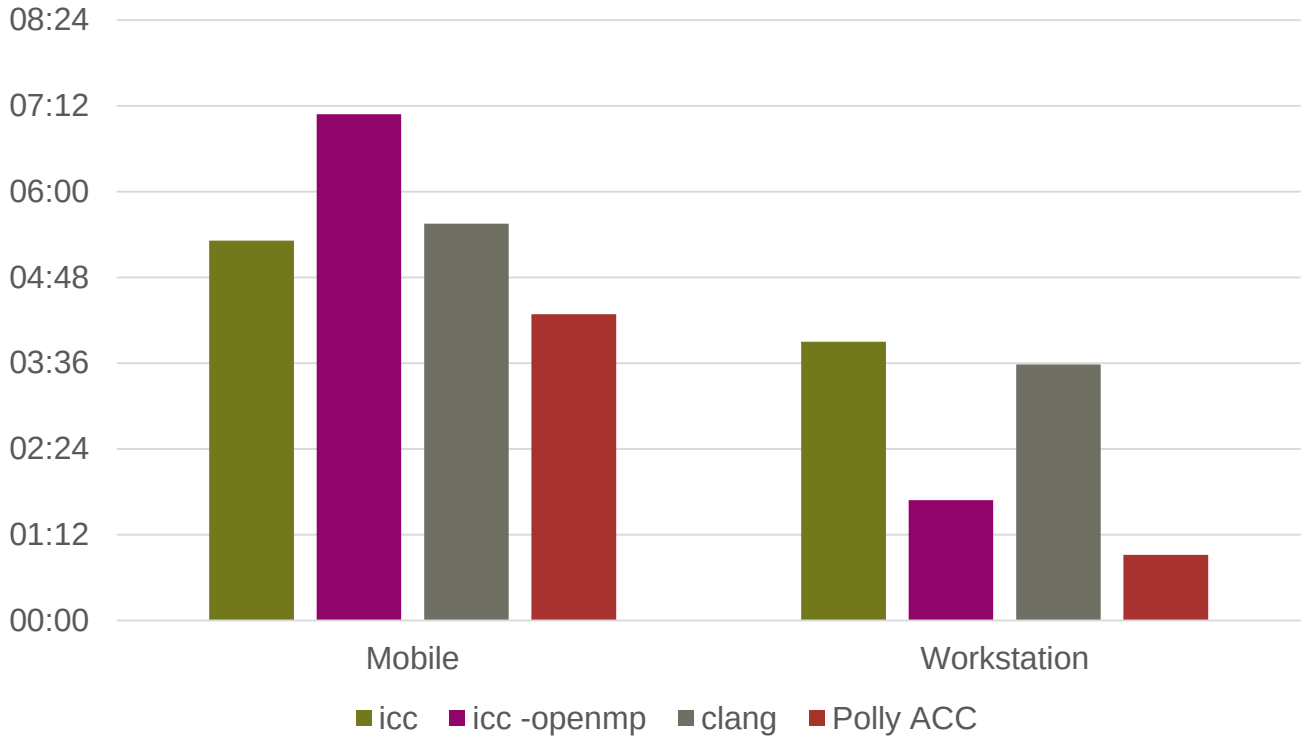


Polybench 3.2



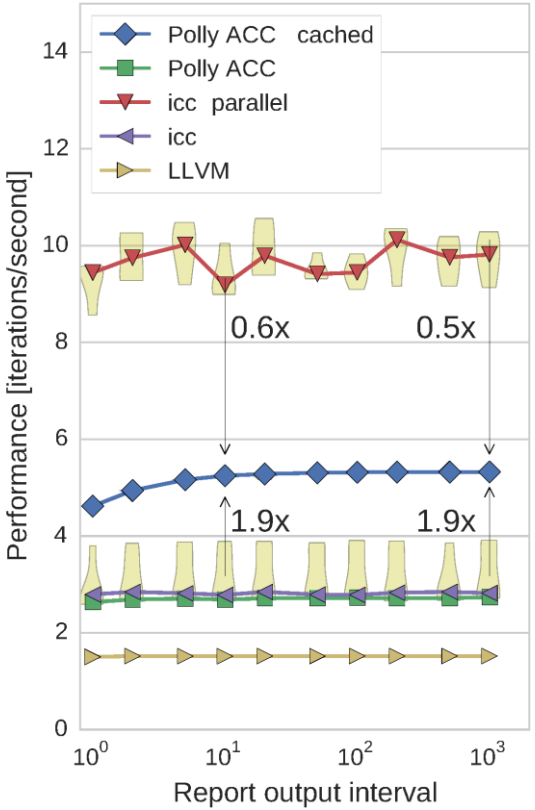
Baseline: icc –O3 (sequential), 10 core CPU + NVIDIA Titan Black (workstation)

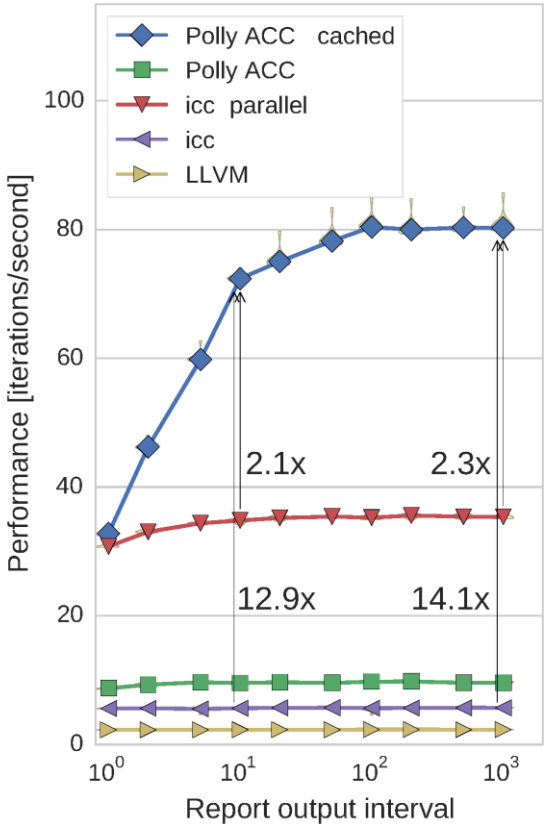
Lattice Boltzmann (SPEC 2006)



Cactus ADM (SPEC 2006)

Mobile

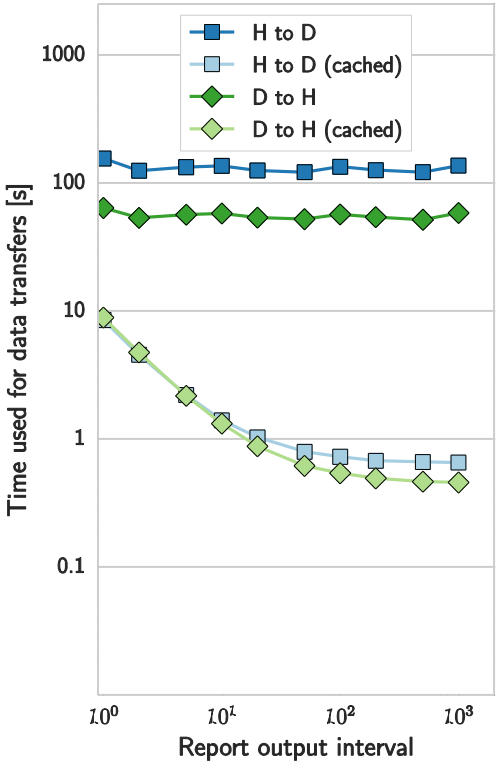




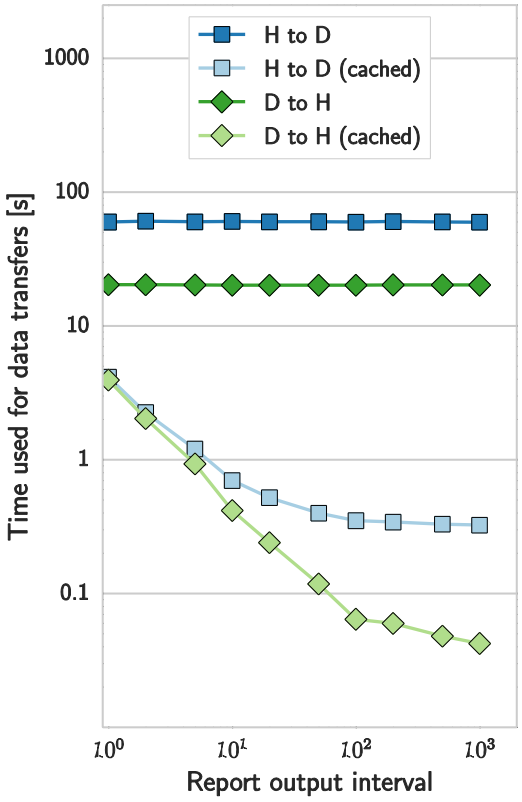
Workstation

Cactus ADM (SPEC 2006) - Data Transfer

Mobile

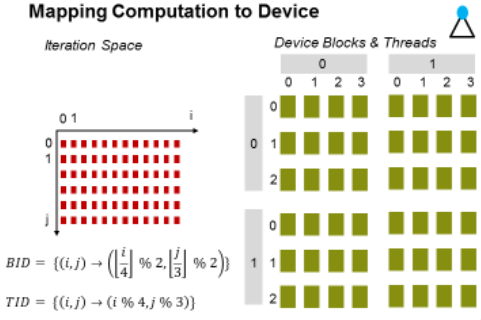


Workstation

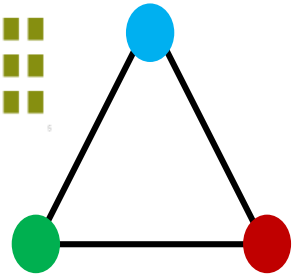


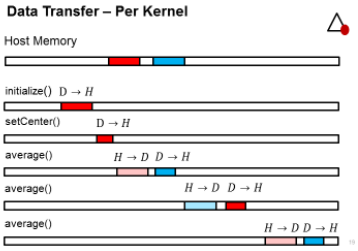
Polly-ACC

http://spcl.inf.ethz.ch/Polly-ACC

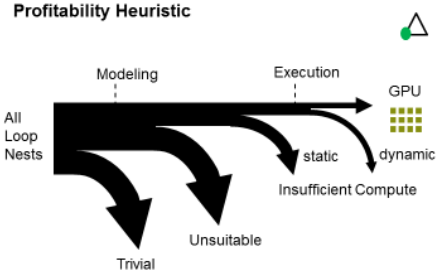


Automatic





"Regression Free"



High Performance