

# PENCIL: Towards a Platform-Neutral Compute Intermediate Language for DSLs

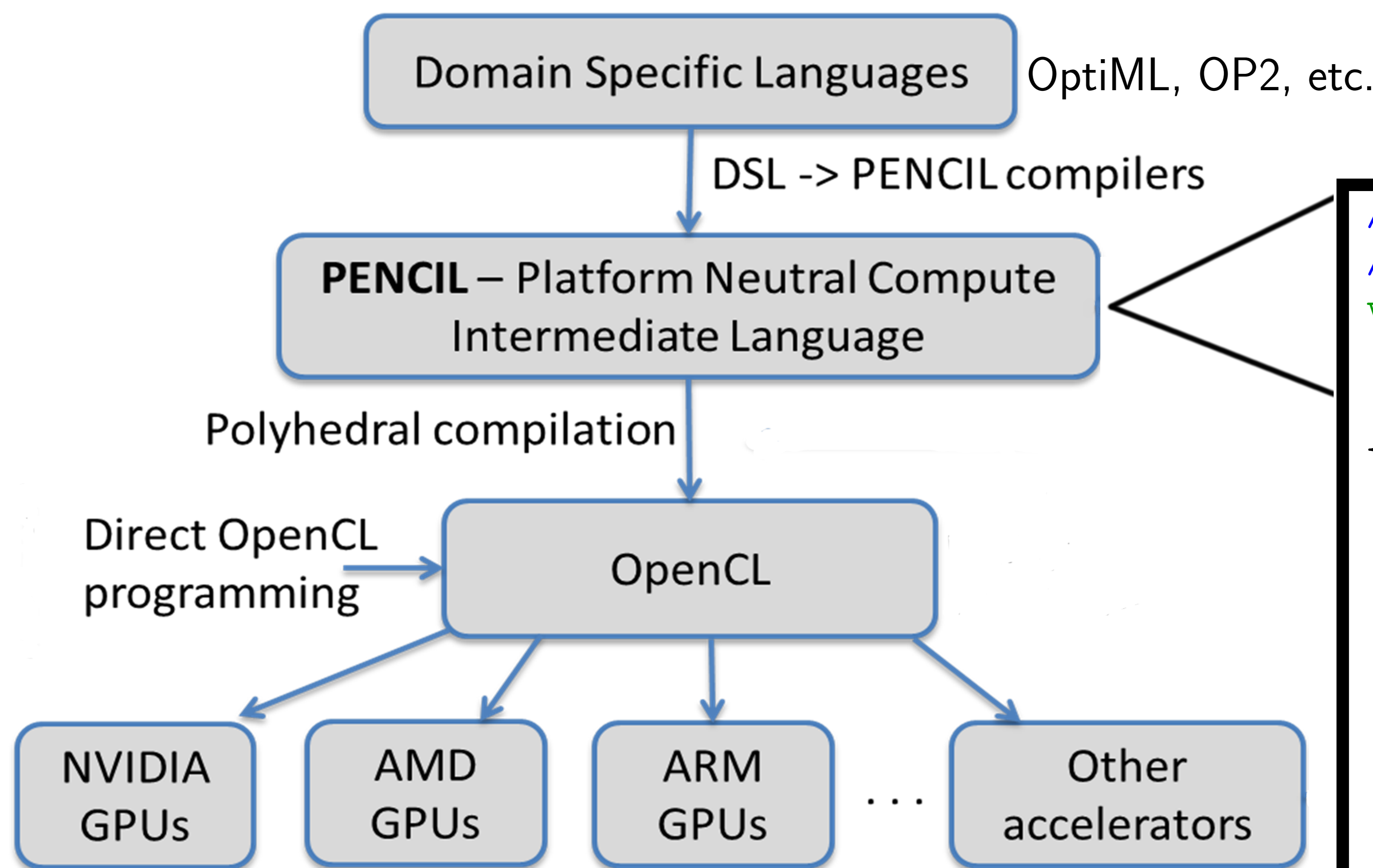
R. Baghdadi A. Cohen S. Guelton S. Verdoolaege J. Inoue T. Grosser  
G. Kouveli A. Kravets A. Lokhmotov C. Nugteren F. Waters A. Donaldson



ARM – Imperial College – ENS/INRIA, Paris

- An intermediate language for DSL compilers and expert developers
- Coding rules, language extensions and directives on top of C
- Assist optimization frameworks by
  - keeping a maximum of information expressed by the DSL
  - eliminating ambiguity for optimizers

## PENCIL-Based Tool Flow



## PENCIL Code Example

```
// Coding rule: no pointers, use arrays
// with restrict (eliminates aliasing).
void f (int N,
        int A[const restrict N][N],
        int C[const restrict N][N])
{
    // polyhedral representation is
    // extracted automatically
    // from static-control loops
    for (int k = 0; k < N; k++)
        for (int j = k; j < N; j++)
            C[i][j] = C[j][i];
    // pragmas to extract polyhedral
    // representation from
    // non-static loops
    #pragma pencil independent
    for (int k = 0; k < N; k++)
        for (int j = 0; j < N; j++)
            A[k][C[j]] = foo(N, A, C);
}

// Summarize external function foo()'s
// data access pattern by a skeleton s().
int foo (int N,
         int A[const restrict N][N],
         int C[const restrict N][N])
    ACCESS(s(n, A, C));

int s (int N,
       int A[const restrict N][N],
       int C[const restrict N][N])
    USE(C); // C is used read-only
    DEF(A[0]); // A is write-only at [0]
}
```

PENCIL provides portable performance and correctness across a range of platforms.

## Implementation

- Starting point: PPCG optimization framework <http://freecode.com/projects/ppcg>
- Input: C for now; PENCIL to be implemented
- Output: CUDA, OpenMP and OpenCL (soon)



FP7 project number 287767