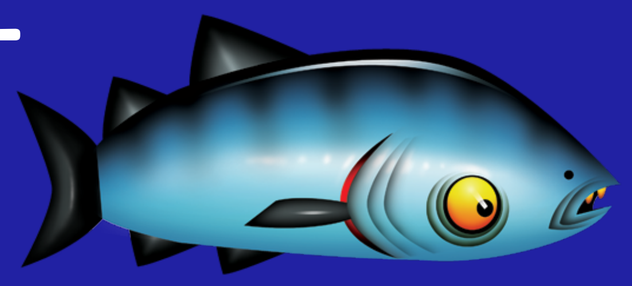
Automatic Compilation of Domain Specific Languages to OpenCL using PENCIL

U. Beaugnon A. Kravets S. van Haastregt D. Tweed J. Absar A. Lokhmotov

ARM, Cambridge; ENS/INRIA, Paris.



http://www.carpproject.eu/

- VOBLA: **V**ehicle for **O**ptimized **B**asic **L**inear **A**lgebra - DSL for programming linear algebra libraries and linear algebra solutions.
- VOBLA compiler generates PENCIL, a domain independent intermediate language (IL).
- PENCIL is designed for efficient mapping to accelerator architectures such as GPUs.
- PENCIL is compiled to efficient, platform-specific OpenCL code using polyhedral techniques.
- Performance of generated OpenCL code on average 2.5X faster than straightforward OpenCL implementations.
- This approach improves programmer productivity and performance portability.

Vobla to PENCIL to OpenCL Compilation Flow

```
function gemm(alpha: Value, in A: SparseIterable<Value>[m][k],
              in B: Value[k][n], beta: Value, out C: Value[m][n])
{
   Cij *= beta forall _, _, Cij in C.sparse;
   C[i][j] += alpha*Ail*B[l][j]
            for i, l, Ail in A.sparse, j in 0:n-1;
}
```

Domain: linear algebra

Matrix-vector multiplication

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{1,1} & 0 & \cdots & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,i} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ \vdots \\ x_n \end{bmatrix}$$

Domain operations captured in DSL

VOBLA

```
sum(Aij*x[j] forall j, Aij in (A[i][*]).sparse)
```

DSL compiler generates PENCIL code

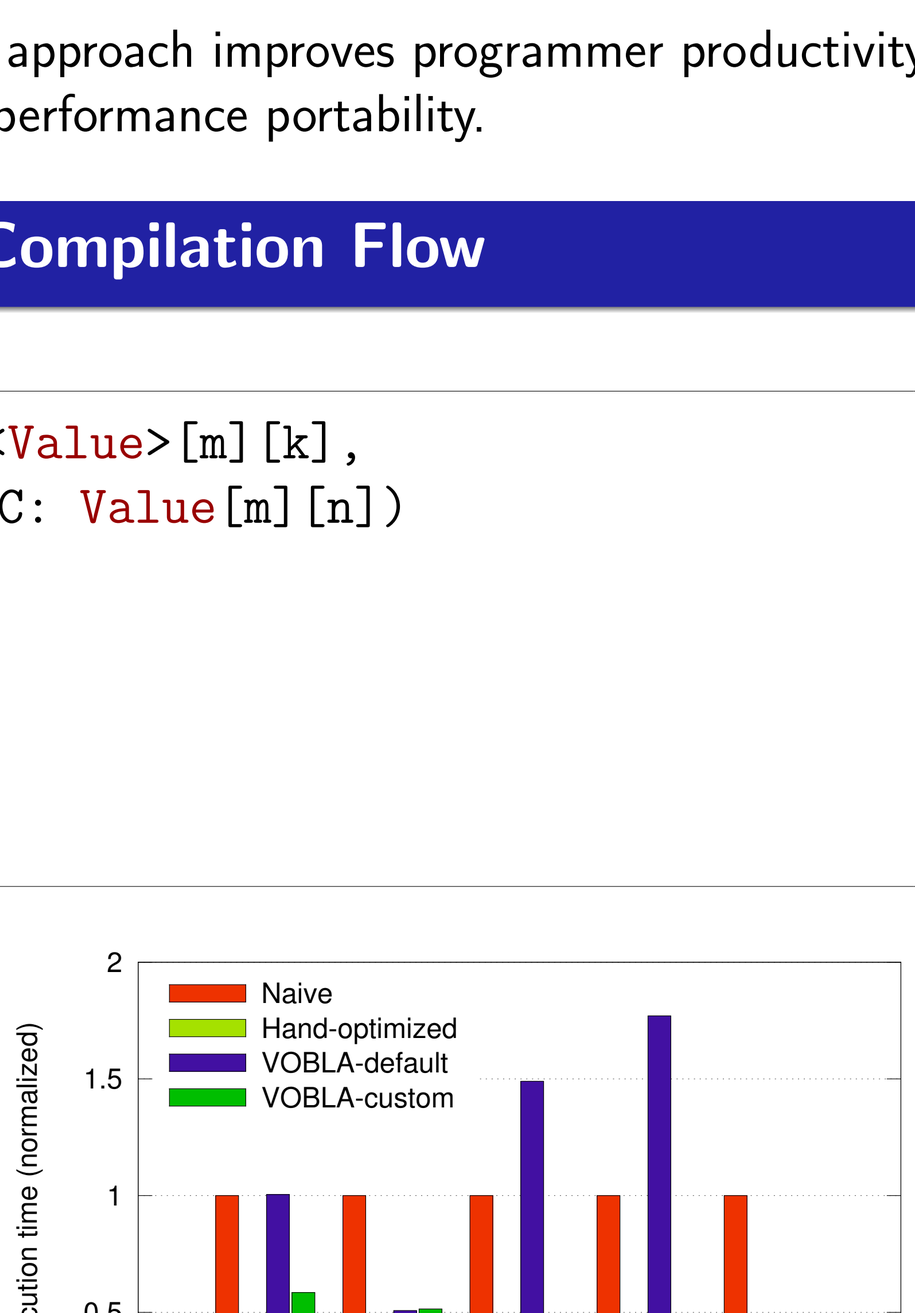
PENCIL

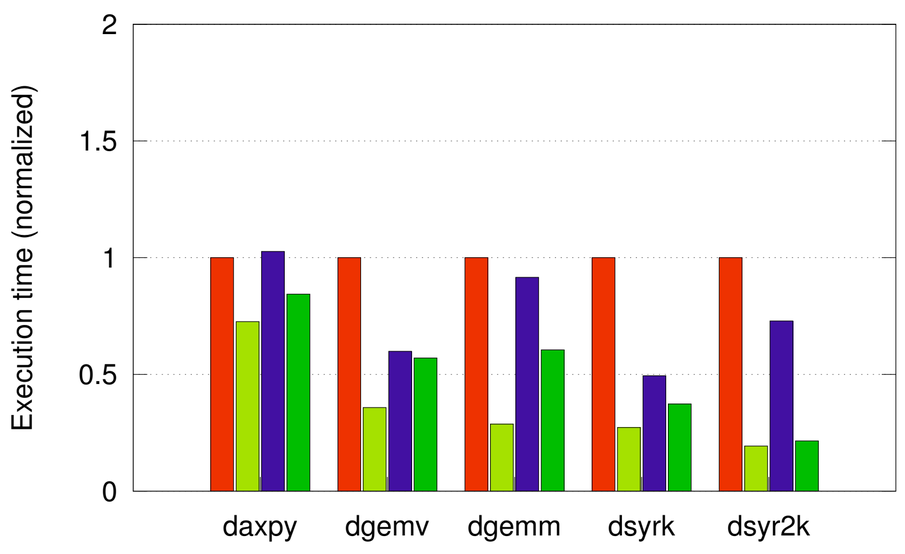
```
acc = 0;
#pragma pencil independent reduction(+: acc)
for (int j = 0; j <= i; j++)
   acc += A[i][j]*x[j];
y[i] = acc;
```

PENCIL compiler generates optimized OpenCL code

OpenCL

```
#pragma OPENCL EXTENSION cl_khr_fp64 : enable
kernel void matvecmul(global double *A, ...) {
  int b0 = get_group_id(0);
  int t0 = get_local_id(0);
  ...
}
```





- R. Baghdadi, A. Cohen, S. Guelton, S. Verdoolaege, J. Inoue, and T. Grosser. PENCIL: Towards a Platform-Neutral Compute Intermediate Language for DSLs. *Workshop on Domain Specific Languages, WOLFHPC'12*, 2012.
- M. Fowler and R. Parsons. *Domain-Specific Languages*. Addison Wesley Publisheres, 2011.
- S. Verdoolaege, J. Carlos Juega, A. Cohen, J. Ignacio Gómez, C. Tenllado, and F. Catthoor. Polyhedral Parallel Code Generation for CUDA. *ACM Trans. Archit. Code Optim.*, 9(4):54:1–54:23, Jan. 2013.

SEVENTH FRAMEWORK PROGRAMME