Motivation
The decline of Dennard scaling has motivated more creative heterogeneous computing architectures with varying types of accelerators such as Graphics Processing Units (GPUs) and reconfigurable logic (FPGAs). However, programming application kernels to take advantage of arbitrary accelerators remains an extremely difficult task.

OpenCL Portability Across GPUs and FPGAs
Jack Wadden and Kevin Skadron

Conclusions
GPU code written in OpenCL is often carefully crafted to take advantage of the architecture, however, this means that its representation may not be optimized for other architectures. We show that the best performing FPGA kernel code is much different in form than the best performing GPU code, thus motivating a higher level abstraction such as a domain specific language for dynamic programming.

Future Work
• Create compilers for recurrence relations to emit high performing code for CPUs/GPUs/FPGAs for arbitrary dynamic programming kernels
• Model input size differences for static scheduling of architecture specific code to the best available accelerator
• Explore other programming paradigms that may perform well on both GPUs and FPGAs and compare performance and portability of these applications and kernels

References

Needleman-Wunsch Dynamic Programming

Needleman-Wunsch is an classic two-dimensional dynamic programming application to calculate global alignments of two string sequences. Starting with two sequences seq1 and seq2, we fill out the scoring table according to the following recurrence relation.

\[ T(i,j) = \begin{cases} \sigma & \text{if } i = 0 \text{ or } j = 0 \\ T(i-1,j-1) + \mathcal{E}(\sigma_i, \sigma_j) & \text{else} \end{cases} \]

Where \( \mathcal{E} \) is an application specific "score" function computed on two characters, \( \sigma_i \) and \( \sigma_j \), and \( T \) gets the maximum of the three values.

Data dependencies can be made to be "causal" meaning that all data can be passed forward in time.

Needleman-Wunsch:

Each colored line represents a diagonal slice of the problem. Problem sizes that are larger than a single OpenCL work-group must be launched as multiple kernels, with each kernel computing a diagonal slice of the problem.