Parallelization of the Smith-Waterman Algorithm with HPX

Stephanie Crillo, 1 Bryce Adelstein-Lelbach, 2,3 Hartmut Kaiser 2,3
1Belmont Abbey College, 2Louisiana State University, 3LSU Center for Computation and Technology

Abstract
Concepts introduced by the Smith-Waterman algorithm are utilized by various bioinformatics systems today, such as BLAST. The algorithm, although accurate, slows greatly when working with larger data sets, due to its $O(n^2)$ complexity 1. We wrote a simple Smith-Waterman algorithm, in both serial and parallel, in an attempt to improve the scalability of Smith-Waterman, as part of our larger effort to improve the scalability of BLAST through parallelization.

Introduction
The Smith-Waterman algorithm, (Figure 1), is designed to find the optimal local alignment of two sequences.

The algorithm builds a matrix based on a specific scoring scheme (Figure 2). In our runs, matches were granted a score of +2, while deletions and insertions (gaps) were granted a score of -1.

Upon completion of the matrix, the algorithm backtracks through the matrix to obtain an optimal alignment (Figure 3).

Strong Scaling of the HPX Smith-Waterman Algorithm
(Significance Length: 4096, Grain Size: 64)

Grain Size Parameter Sweep
(Significance Length: 512, OS-Threads: 22)

Results
Parallelization of the Smith-Waterman code was done by dividing the matrix into smaller submatrices and computing the scores of these submatrices in parallel where possible (see Figure 4 and 5). Additionally, backtracking of each of these components was computed as the matrix was being constructed.

References

Acknowledgments
1. NSF Grants 1240655, 1117479, 1048019 and 1029161
2. DOE ASCR X-Book Program
3. Center for Computation and Technology