

CS 388 Final Project Proposal

Aaron Stacy

Calvin MacKenzie

We propose implementing a parallel version of the Viterbi algorithm and characterizing its performance over a range of workloads. We plan to test on the Wall Street Journal dataset from the Penn Treebank and vary the size of the dataset to measure performance. We plan to run the algorithm on the TACC cluster to find both strong and weak scaling¹ characteristics.

There are 3 levels of parallelism we believe we can exploit in running the Viterbi algorithm on a significant dataset:

1. Parallelizing across sentences

When calculating either sequence labels or observation probability, each sentence is considered independently. This presents an easy opportunity to parallelize, since each calculation is independent of the other. In the case of observation probability, the results can be combined via a parallel reduction.

2. Parallelizing within sequence steps

The Viterbi algorithm performs a $O(n^2)$ calculation for each item in the sequence. This can be viewed as a matrix-vector multiplication with a vector of 1's, so applying a parallel matrix-vector routine, such as one of those found in Intel's Thread Building Blocks² library should provide performance benefits.

3. Parallelizing across sequence items

The dynamic programming algorithm Viterbi uses requires each step to use the previous step's value to calculate. The recursion is as follows:

$$v_{i,j} = b_j(i) \sum_{k=0}^n v_{i-1,k} \cdot a_{k,j} \quad (1)$$

Although this calculation cannot be parallelized across sequence steps as-is, it should be possible if the recursion can be converted to an associative operation. If so, then a parallel scan³ algorithm can be used to reduce the $O(n)$ operations to $O(\log n)$.

¹http://en.wikipedia.org/wiki/Scalability#Weak_versus_strong_scaling

²http://www.threadingbuildingblocks.org/docs/help/reference/algorithms/range_concept/blocked_range2d_cls.htm

³http://en.wikipedia.org/wiki/Prefix_sum#Parallel_algorithm