CPSC 221H – Example Lab Report

Introduction

This report serves as an example lab report for CPSC 221H. This example details the experimental setup and performance results for scalability tests using the Standard Template Library algorithm Accumulate. The algorithm accumulates a range of elements specified by two Input Iterators, and returns the final result.

Theoretical Analysis

The complexity of STL Accumulate is linear, O(n) where n is the input size. A single linear traversal is needed to accumulate all requested elements.

Experimental Setup

The experiments described in this report were performed on a Linux computer with a P4 2.8Ghz with 1Gb RAM running CentOS 5.2. C++ code was compiled using GCC 4.1.2 and –O3 optimizations.

The standard library ctime was used for timing the experiments, and a minimum of 10 executions averaged. For small input sizes, hundreds of thousands of executions were averaged to compensate for the clock resolution.

To test STL Accumulate, a vector of doubles was used as input initialized with random values. The size of the vector ranges from 2 to 67108864 elements. This range show performance at a very small scale, and also when using almost all available free memory on the system. Tests with larger inputs that require virtual memory swapping are not shown.

Experimental Results

Figure 1 shows the running time of STL Accumulate vs the input size on a log-log plot. The algorithm scales close to linearly as expected.

Figure 2 shows Time/size vs. size on a log-x plot. We can use this plot to find the Big-O constants from the equation: $0 <= f(n) <= c*g(n)$ for all $n >= n0$. We can see that the Big-O constant in terms of seconds is 2.5e-8 where n0 is 8. The red and blue arrows mark these respectively.

Figure 3 shows the Big-oh bounding line represented by 2.5e-8 * n over the running time of STL Accumulate. Also, n0=8 can be seen on this plot.

Summary

This report shows an experimental performance evaluation of STL Accumulate. Its O(n) complexity is demonstrated through experimental tests, with asymptotic constants and n0 shown.

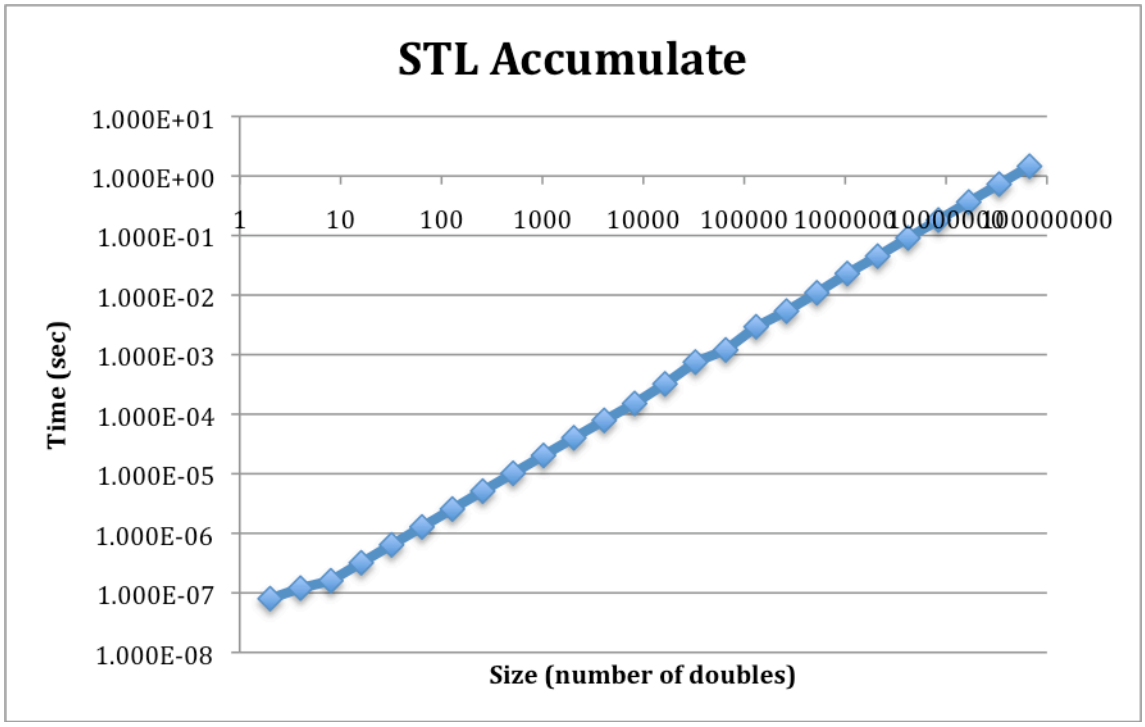

Figure 1: Execution time of STL Accumulate vs. input size on a log-log plot.

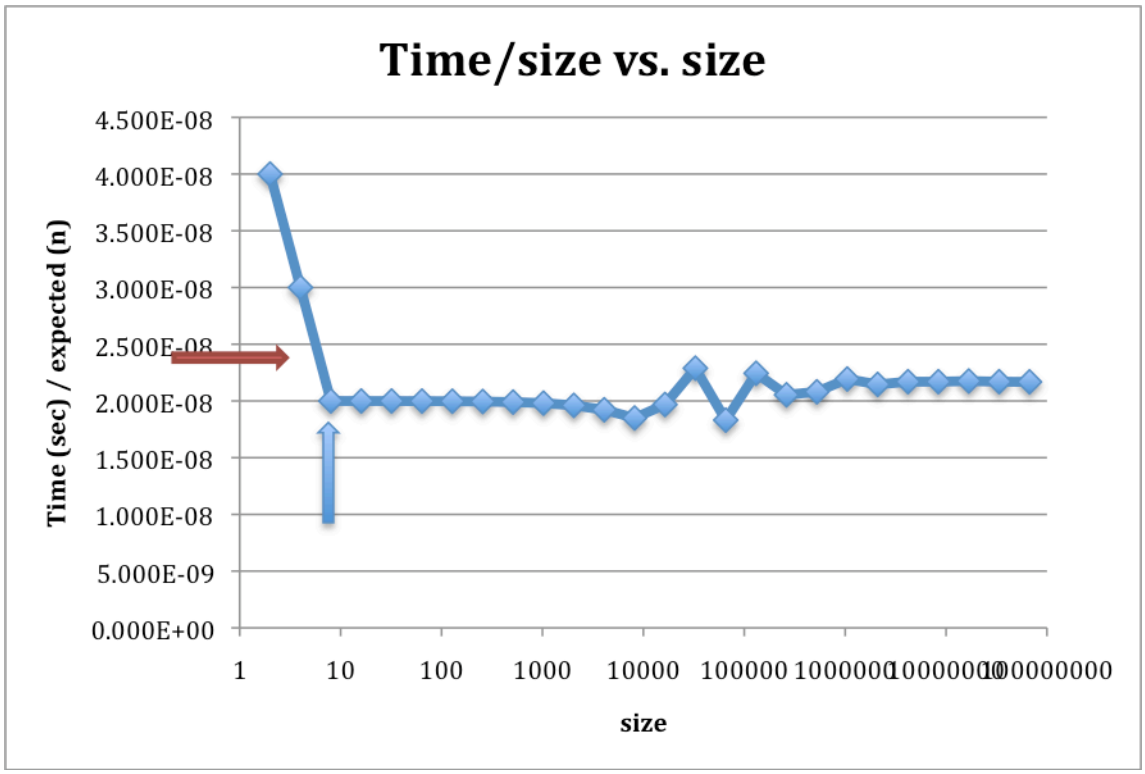

Figure 2: Time/size vs. size for STL Accumulate on a log-x plot.

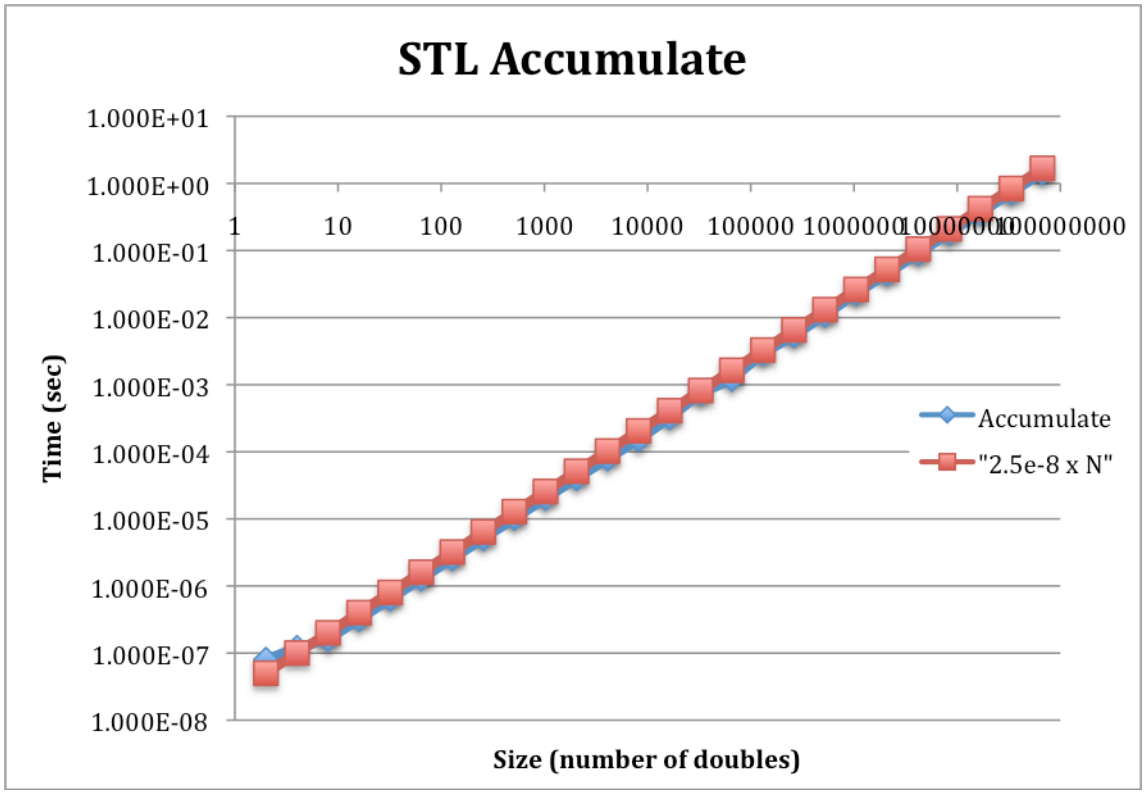

Figure 3: Execution time of STL Accumulate vs. input size on a log-log plot. Also shown is 2.5e-8 * size shown as a bounding line.