Introduction

In C++, the standard library contains link and vector classes. The link class uses a doubly-linked list data structure; whereas the vector class utilizes an array with growth by a factor. Below is provided a graph comparing the number of pushes and runtime for each class.

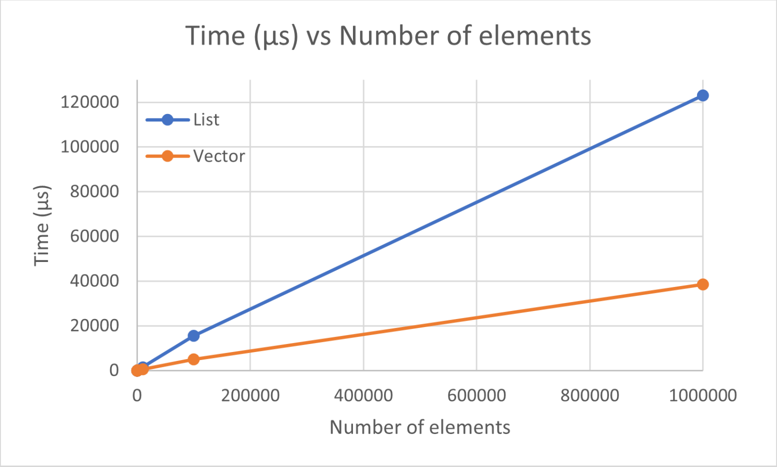

FIGURE 1. The runtime (microseconds) versus the number of push operations

Theoretical Time Complexity (push operation)

For the vector class, the array data structure (growth by a factor) has a time complexity for its push operation as O(1). Its worst-case is O(n). This is because resizing an array requires reallocating all memory and copying the old array values to the new memory. Its amortized over the n spaces created; thus, giving an average complexity of O(1).

For the list class, the time complexity for the push back operation is O(1). This is because we simply create another pointer and add the element to the list regardless of the size of the list.

Principle of Spacial Locality

The processor contains multiple levels of memory storage. Cachegrind outputs information relating to the first level of the instructional and data caches, and the last level cache. The first level is the smallest and closest to the processor; thus, one can expect the retrieval of data stored here to be quickest. The cache takes advantage of temporal locality by retrieving contiguous memory. As arrays store their values contiguous in memory, this is why the runtime of the vector class is less than the list class.

The list Cachegrind output has significantly more last level misses than that of the vector class. The L3 (last level) misses for the list class is ∼1,000,000; whereas it is ∼150,000 for the vector class. Cache misses are less efficient than hits (naturally as it must search further), especially on the last level since it now has to access the main memory. The vector class (array) stores contiguous memory. Because the cache retrieves contiguous memory, there are significantly less misses for the vector class. The linked list data structure is non-contiguous storage and this is why the cache must search more for the location of its data.