

3.1 VOLATILE Counter

When we use the volatile counter variant the result will be false. The cause for this is that while every thread is fetching the value of the counter from the storage and not from the cache this fetching can be still concurrent and appear simultaneously. Therefore several can have the same value for the counter and increment this value and write it back. Therefore we get a falsely counter.

3.2 ATOMIC Counter

When we use the AtomicInteger's CompareAndSet(expectedValue, updatedValue)-method it will return TRUE if the expected value was the actual value and FALSE if the expected value was wrong. This can be used in a while loop to call this method until it has worked. Therefore the right value for the counter is returned at the end. But a disadvantage of this method is that it can take a very long time for all the threads to terminate, because it is not clear how often the CompareAndSet method is returning FALSE.