

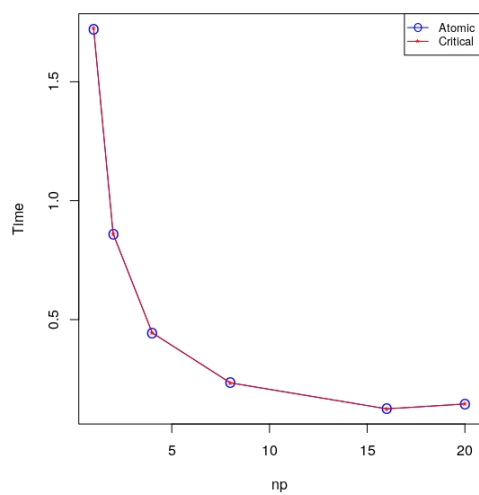
EXERCISE 1 - APPROXIMATE π

The exercise requires to compute π using the midpoint rule. In the code both the serial and the parallel method are implemented. For the parallel method an OpenMP approach is used and in particular 3 different pragma sections are used (in order to protect the summation variable):

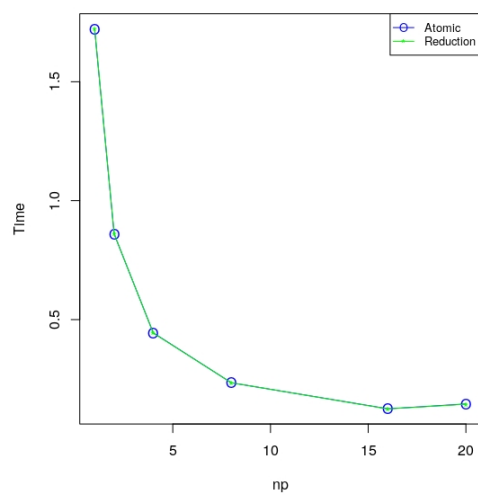
- *Atomic*: with this directive mutual exclusion for some simple operations is enabled and these are converted into special hardware instructions. This directive allows access to a specific memory location atomically, it ensures that race condition are avoided through direct control of concurrent threads.
- *Critical*: this directive ensures that threads have mutually exclusive access to a block of code, only one thread can enter a critical section, so it effectively serializes the execution of this block of code.
- *Reduction*: with this clause a private variable for each is created on which each of them work and at the end all threads are accumulated using the operator (specified in the directive).

The 3 methods are compared observing the time of execution and the code is tested using 1, 2, 4, 8, 16, 20 threads (using the *pi.sh* script in this folder).

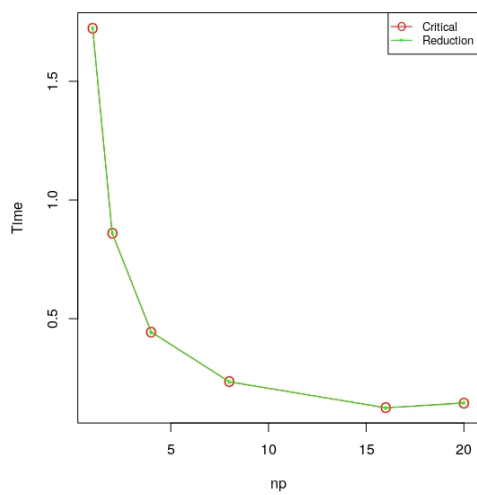
The results are reported in the *res.txt* file and summarized in the plots below and we can notice that they behave very similar.



(a) Atomic vs Critical



(b) Atomic vs Reduction



(c) Critical vs Reduction

Figure 1: Comparison of the 3 different pragmas