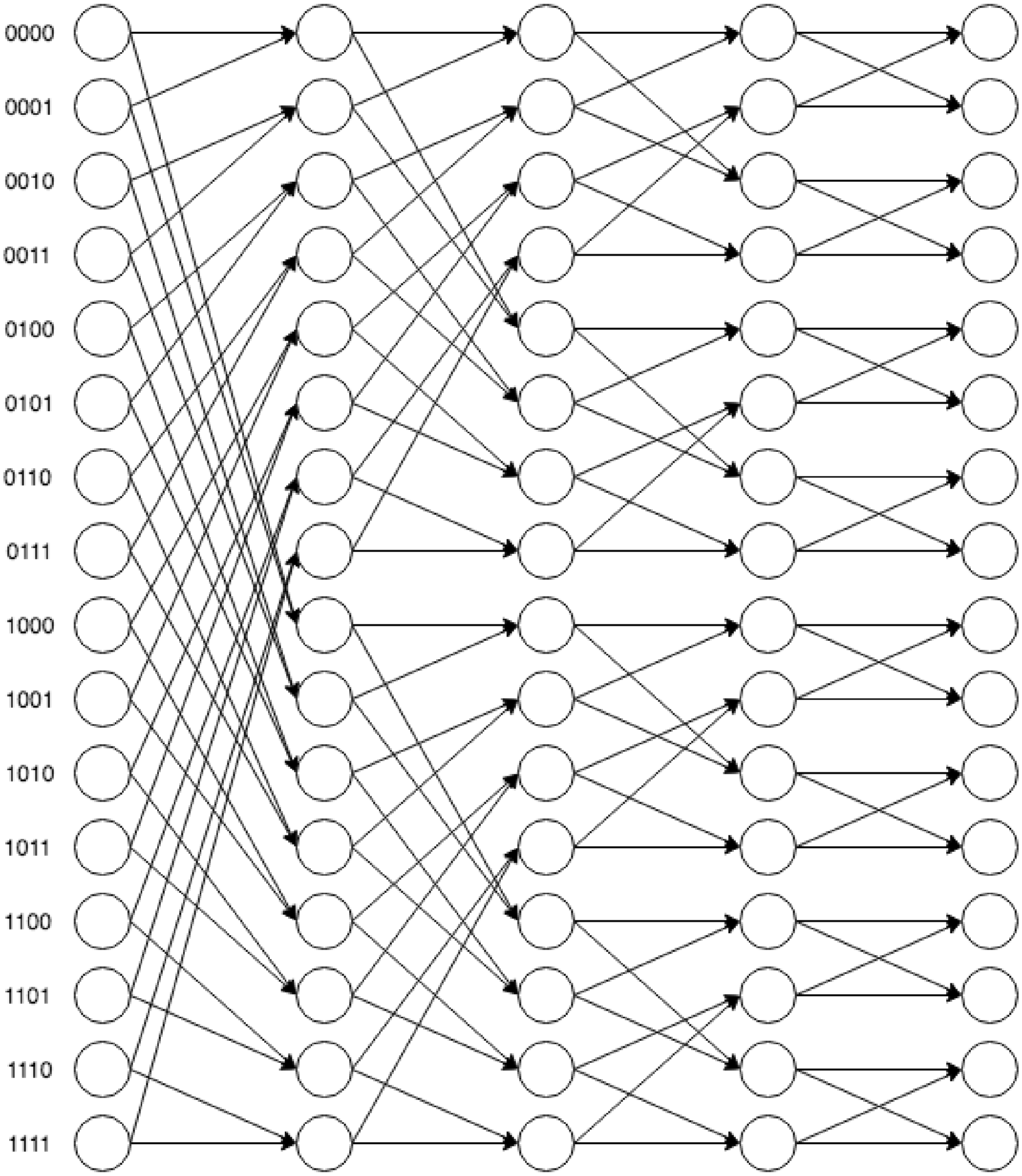
1. Uniform Memory Access (UMA) and Non Uniform Memory Access (NUMA) systems are both types of shared memory computing systems. In a NUMA system each node is connected to its own personal main memory block, one node may access another's memory only through the logical interconnect between the chips. This allows each chip fast access to its individual main memory but to reach data located on that of another node in the system takes much longer. In a UMA system all nodes share a single large block of main memory. A bus or multistage interconnect allows all nodes to take a similar amount of time to access any point in main memory. For all nodes to reach all points in memory the interconnect requires some logical complexity which renders an access time slower than a NUMA chip's access to it's local block, but faster than the time it would take for a NUMA chip to access memory across the system.

2.



3. Atomic parallel execution of the trapezoid accumulation algorithm results in a drastic decrease in performance compared to a simple sequential execution. This is due to the overhead of sharing one resource between all threads for summation of the result, each thread must wait it’s turn at each iteration to get access to the atomic variable. For the reduction technique the observed speedup is up to 7x, each thread can independently iterate through it’s section of the pi approximation and store the intermediate result in a local variable, at the end the local variables are summed by the reduction for the end result. This technique involves no waiting time except for the small overhead of allocating extra local variables and combining the temporary results, the processing capacity of the multiple threads are taken advantage of rather than splitting up the logic only to wait on memory access like in the atomic method. The overhead dominates for small iteration counts or small numbers of threads where the leverage of multiprocessing is smaller but speedup generally increases with both problem size and number of threads.