Quiz 2 Solution

1. Consider adding hardware multithreading to a core. What is the impact on throughput as measured by instruction per cycle, and what is the impact on latency as measured by the time a thread can finish a certain number of instructions?

   Typically, hardware-implemented threads are added to keep the processor busy despite cache and TLB misses. They share the same pipeline, cache and memory bandwidth. Because of this, if the code is well behaved (good locality of references) then the threads will interfere with one another and there will be some benefit from exploiting pipeline bubbles. But the competition will likely slow down each thread than if it were running all by itself. Therefore the throughput will improve in general, but the latency per thread will deteriorate.

2. Consider Amdahl's law. There is some code that consists of a sequential part and then a recursive function. How do you apply the law in this case?

   A recursive function can be converted into a loop and vice versa. However, compilers often find it difficult to deal with recursive functions. But if the equivalent loop to a recursive function can be parallelized, then the recursive function can also be parallelized, and the law will apply in this case to express a limit on the improvement that can be obtained by parallelizing the recursive function.

3. A system shows utilization of 60% at the network interface, 60% in the memory bus, and 15% at the CPU. If you increase the capacity of the network and memory, what would be the impact on performance?

   It is of course difficult to answer this problem with 100% certainty, because the relationship between the utilizations cannot be discerned from the information. Nevertheless, none of the components in the described system appears to be a bottleneck, therefore there may not be much, if any, performance improvement by increasing the capacity of either the memory or the network.