**Part 2 Questions – Zihan Wan**

**1. What is the FLOPS? and usage in parallel computing?**

FLOPS is a measure of computer performance, useful in fields of scientific computations that require floating-point calculations. Using floating-point encoding, extremely long numbers can be handled relatively easily.

**2. Simply explain the computer architecture of von Neumann.**

von Neumann architecture is known as 'stored-program computer' -both program instructions and data are kept in electronic memory. It is comprised of four main components: Memory, Control Unit, Arithmetic Logic Unit, Input/Output. Control unit fetches instructions/data from memory, decodes the instructions and then sequentially coordinates operations to accomplish the programmed task. Arithmetic Unit performs basic arithmetic operations. Input/Output is the interface to the human operator.

**3. Simply explain: what is shared memory architecture and what is distributed memory architecture?**

Shared memory architecture provides a virtual address area shared between any or all nodes, all processors have the ability to access all memory as global address space. Distributed memory architecture needs a communication network to connect inter-processor memory, and it requires explicit commands to transfer data from one processing element to another.

**4. List three factors that can cause parallel overhead and simply explain.**

- Task start-up time: Extensive additional calculations increase start-up time.

- Synchronizations: Additional resources are required for the synchronization of processes (threads).

- Data communications: Mainly determined by latency and bandwidth.

**5. Give the formula of Amdahl's Law including the number of processors.**

speedup = 1/(P/N+S), where P = parallel fraction, N= number of processors and S = serial fraction

**6. Please give the parallel programming models in common use and simply explain.**

- Shared Memory Model: Processes share a common address space, which they read and write to asynchronously.

- Distributed Memory Model: A set of tasks that use their own local memory during computation. Multiple tasks can reside on the same physical machine and/or across an arbitrary number of machines.

- Data Parallel Model: Address space is treated globally. A set of tasks work collectively on the same data structure, however, each task works on a different partition of the same data structure.

- Hybrid Model: Combines more than one of the previously described programming models. It lends itself well to the most popular (currently) hardware environment of clustered multi/many-core machines.

- Single Program Multiple Data(SPMD): Single Program - All tasks execute their copy of the same program simultaneously. Multiple Data - All tasks may use different data.

- Multiple Program Multiple Data(MPMD): Multiple Program - Tasks may execute different programs simultaneously. Multiple Data - All tasks may use different data.

**7. Simply give 3 possibilities for poor parallel performance and simply explain.**

- Load balance, where the computing resources aren't contributing equally to the problem.

- Network effects, such as latency and bandwidth, that impact performance and communication overhead.

- Impacts from use of RAM and virtual memory, such as cache misses and page faults.

**8. Briefly explain the hybrid parallel programming model on current supercomputers and HPC clusters.**

Currently, a common example of a hybrid model is the combination of the message passing model (MPI) with the threads model (OpenMP). Another similar and increasingly popular example of a hybrid model is using MPI with CPU-GPU (graphics processing unit) programming.

**9. Please list the three primary API components in OpenMP and give simple examples.**

- Compiler Directives: \#pragma omp parallel default(shared) private(beta,pi)

- Runtime Library Routines:

#include <omp.h>

int omp\_get\_num\_threads(void)

- Environment Variables: export OMP\_NUM\_THREADS=8

**10. Please write the steps to use gcc compiler to compile a C file “hello.c” with OpenMP flag; set the environment variable of 4 threads on the Linux operating system by using “export”; and run this executable. Note: only give the commands.**

$ gcc -o helloc -fopenmp hello.c

$ export OMP\_NUM\_THREADS=4

$ ./helloc