Question 1  [18 marks] General

(a) In the context of parallel systems, provide concise definitions for each of the following:

(i) Super-linear speedup
(ii) Race condition
(iii) Atomic operation

[6 marks]

(b) Domain decomposition and pipelining are two approaches to parallelization. Describe how you would parallelize the sorting of \( n \) integers using \( p \) processors taking:

(i) a domain decomposition approach
(ii) a pipelining approach (hint: consider insertion sort)

[6 marks]

(c) You have two parallel codes, A and B, that you run on the same machine. Code A takes 41 minutes on 4 processors and 32 minutes on 10 processors. Code B takes 22 minutes on 5 processors and 16 minutes on 8 processors.

(i) Use Amdahl’s law to predict the execution time of each code on 1 processor of this machine. Which takes the shortest time?
(ii) Use Amdahl’s law to predict the execution time of each code on 100 processors.
(iii) From the above you have time estimates for codes A and B on 1 and 100 processors. For each estimate state whether you think it will be an underestimate or an overestimate and why.

[6 marks]
Question 2  [18 marks]  Message Passing

(a) In the language of MPI, what is being described in each of the following cases?

(i) A send routine that does not return until the send buffer can be reused
(ii) A send routine that does not return until receipt of the message at its destination has been acknowledged
(iii) An operation in which one process sends the same data to several others
(iv) An operation in which one process distributes different elements of a local array to several others
(v) An operation that obtains data from a remote processor without the application code running at the remote node having to explicitly issue a send call

(b) Explain what the following MPI program is doing, and state the output that is printed if the code is run using 7 MPI processes:

```c
int main (int argc, char *argv[]) {
    int token, NP, myrank;
    MPI_Status status;
    MPI_Init (&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &NP);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
    if (myrank != 0) {
        MPI_Recv(&token, 1, MPI_INT, myrank - 1, 0, MPI_COMM_WORLD, &status);
    } else {
        token = -1;
    }
    token += 2;
    MPI_Send(&token, 1, MPI_INT, (myrank + 1) % NP, 0, MPI_COMM_WORLD);
    if (myrank == 0) {
        MPI_Recv(&token, 1, MPI_INT, NP - 1, 0, MPI_COMM_WORLD, &status);
    }
    printf("rank and token %d %d\n", myrank, token);
    MPI_Finalize();
}
```

(c) Write an MPI program that implements a workpool consisting of a master process and 5 slave processes. The master sends a task consisting of two random integers to each slave. The slaves adds the two integers and return the result to the master. The master then sends another task to the slave, again consisting of two random integers to be added together. This continues until 100 tasks are completed. Your program does not need to be syntactically correct, but it should give sufficient detail to indicate your intent.

[5 marks]
Question 3 [18 marks] Shared Memory

(a) Parallelize the following code using openMP pragmas. Assume that the target machine has a cache line size of 128B, that the size of an int is 4B, and the arrays contain ints. Be sure to explicitly specify the “schedule” options that should be used, even if you want to use the default options. For each please rewrite as much code as necessary to make your intent clear. If necessary you can assume that the variable P represents the number of processors to be used. Assume that N is large (in the tens of thousands or more). You must explicitly list all variables within the range of a parallel pragma that are private using the private() directive.

(i) for (i=0; i<N; i++){
    for (j=0; j<N; j++){
    }
}

(ii) C[0] = 1;
    for (i=1; i<N; i++){
        C[i] = C[i-1];
        for (j=0; j<N; j++){
            C[i] *= A[i*N+j] + B[i*N+j];
        }
    }

(iii) typedef struct element
    { int value;
      struct element *next;
    } Element;
    Element *D[N]; // array of pointers to linked lists of varying length
    int C[N];
    ...
    for (i=0; i<N; i++){
        C[i] = computeAverageValueOfAllListElements(D[i]);
    }

[9 marks]

(b) Three processors P0, P1, and P2 perform the following read (r) and write (w) operations on shared integer variables x and y that both initially have values of 0. For example r(x)2 means variable x is read from memory and found to have a value of 2.

<table>
<thead>
<tr>
<th>Processor</th>
<th>First to last event list</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>w(x)1 r(x)2 r(y)3</td>
</tr>
<tr>
<td>P1</td>
<td>r(x)1 w(y)3 r(y)1</td>
</tr>
<tr>
<td>P2</td>
<td>r(y)3 w(x)2 w(y)1</td>
</tr>
</tbody>
</table>

(i) What are the two requirements for sequential memory consistency?
(ii) Show EITHER that the above events can be ordered in a sequentially consistent manner or explain why they cannot.
(iii) Most modern shared memory computers do NOT implement sequential memory consistency, Explain why this is the case and detail two aspects of the hardware that may lead to a loss of sequential memory consistency.

[9 marks]
Question 4 [18 marks] GPU Programming

(a) Give two examples of applications or computational kernels that are likely to perform well on a GPU system, and two examples of applications or computational kernels that are likely to perform poorly. Provide brief justifications for each of your choices. [4 marks]

(b) Consider the following CUDA host and kernel code for vector addition

```c
__global__
void vecAddKernel(float* A_d, float* B_d, float* C_d, int n)
{
    int i = threadIdx.x + blockDim.x * blockIdx.x;
    if(i<n) C_d[i] = A_d[i] + B_d[i];
}

int vectAdd(float* A, float* B, float* C, int n)
{
    //assume that size has been set to the actual length of arrays A, B, and C
    int size = n * sizeof(float);
    int size = n * sizeof(float);
    cudaMalloc((void**) &A_d, size);
    cudaMalloc((void**) &B_d, size);
    cudaMalloc((void**) &C_d, size);
    cudaMemcpy(A_d, A, size, cudaMemcpyHostToDevice);
    cudaMemcpy(B_d, B, size, cudaMemcpyHostToDevice);
    vecAddKernel<<<ceil(n/256), 256>>>(A_d, B_d, C_d, n);
    cudaMemcpy(C, C_d, size, cudaMemcpyDeviceToHost);
}
```

(i) Assume that the size of A, B, and C is 1000 elements. How many thread blocks will be generated? How many warps are there in each thread block? How many threads will be created in the entire grid?

(ii) In the context of GPU computing what is SIMT? How does this differ from SIMD?

(iii) This code exhibits control/warp divergence for n = 1000, but not for n = 768. What is control/warp divergence and why is there a difference in behavior between these two array sizes?

(iv) Data structure padding can be used to eliminate control/warp divergence. How would you modify this code to use padding to eliminate control/warp divergence?

(v) How do you expect the above code to perform compared to just using the host CPU. Consider performance as a function of the vector size. [14 marks]
Question 5 [18 marks] Parallel Input/Output

(a) Using the diagram below, answer the following questions about the Lustre filesystem:

(i) What is the role of the Management Servers (MGSs)? Outline how they interact with the other components of the Lustre filesystem.

(ii) What is the role of the Metadata Servers (MDSs) in Lustre? Contrast their role with that of the Name Node in the Hadoop filesystem (HDFS).

(iii) The Hadoop filesystem is not POSIX compliant. What was the reason for not supporting POSIX semantics in HDFS? List one aspect of the POSIX filesystem standard that is not supported by HDFS.

(iv) Why are pairs of servers used for OSSs, MDSs and MGSs?

(v) What is an OST? How does Lustre ensure data-availability in the face of individual disk failure at the OST level? How does HDFS provide protection against disk failure at a particular data node?

(vi) Define what the stripe width and stripe count are in the context of a Lustre filesystem. If checkpoint32.nc is written by Lustre client 3, what is the stripe count for objects H01 through H04 allocated on OSSs 1, 2, 3 and 4. Assuming Lustre default values for stripe width, what is the filesize for checkpoint32.nc?

(vii) If an HDFS was constructed using 1000 6TB disks, approximately how much disk space would be available to the user? How would this change if you used the same disks to construct a Lustre filesystem? Explain your answer and any assumptions you make.

[12 marks]
(b) Consider the following two tables, R(u,b) and S(u,c), where u is a unique student identifier and b and c are the marks that the students obtain in their biology and chemistry courses respectively.

<table>
<thead>
<tr>
<th>Table R(u,b)</th>
<th>Table S(u,c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>u (ID) b (Mark_Bio)</td>
<td>u (ID) c (Mark_Chem)</td>
</tr>
<tr>
<td>u12345 83</td>
<td>u12342 67</td>
</tr>
<tr>
<td>u12346 37</td>
<td>u12345 29</td>
</tr>
<tr>
<td>u12347 54</td>
<td>u12347 87</td>
</tr>
<tr>
<td>u12349 71</td>
<td>u12348 11</td>
</tr>
<tr>
<td></td>
<td>u12349 67</td>
</tr>
<tr>
<td></td>
<td>u12358 92</td>
</tr>
</tbody>
</table>

Explain how you could use the map/reduce framework to combine the data from the two tables into a new table, T(u,b,c), containing the biology and chemistry marks (as applicable) for each unique student identifier. Your explanation should include a description of the map and reduce functions you would write.

[6 marks]