CS601: Software Development for Scientific Computing

Maximum Points: 25 | Mid-semester examination | 20/09/2023, 9:30AM to 11:30AM

Instructions: This exam has two parts. Part I is open-book, open-notes (printed/written). No electronic devices allowed. Part II is take-home. The submission instructions for part II are the same as in programming assignment 1. State your assumptions (if any) clearly.

Part I **(19 points)**:

1. List all the ideas that you have learnt in CS601 to implement a fast matrix-multiplication kernel. **(2 points)**

2.

```
1  CXX=g++
2  CFLAGS=-I./inc
3  ifeq ($(DEBUG), 1)
4  CFLAGS += -g
5  endif
6
7  Solution: RDomain.o GridFn.o Main.cpp
8          $(CXX) $(CFLAGS) $^ -o clean
9  RDomain.o: RDomain.cpp
10         $(CXX) $(CFLAGS) -c $< -o $@
11 GridFn.o: GridFn.cpp
12         $(CXX) $(CFLAGS) -c $< -o $@
13 clean:
14         rm -f *.o clean
```

a. A `makefile` is shown in the figure. When you run the `make` command, 3 commands are executed. Write the commands in the exact order in which they are executed (you may just write the line number containing the command. Assume that the program has no syntax error or warnings). **(1.5 points)**

b. After executing `make` command, you edit `GridFn.cpp` and again execute the `make` command. What are all the commands that are executed? Write line numbers only. **(1 point)**

c. Next, what happens when you execute the `make clean` command? **(1 point)**

d. How would you trigger the command shown in line 14? **(1.5 points)**

3.

```
#include<iostream>
#include<cstdlib>
int fact(int n){
 if(n<0)
    return -1;
 if(n==0)
    return 1;
 return n*fact(n-1);
}

int main(int argc, char* argv[])
 if(argc!=2)
        exit(-1);
 int input=atoi(argv[1]);
 std::cout<<fact(5);
}
```

a. Consider the program below. You wish to start a debug session using GDB and pass a command line argument of 5. What is the command that you need to execute to start the debug session? You can assume that the name of the target is `a.out` **(2 points)**

b. You wish to insert a breakpoint at the beginning of the `fact` function. What is the gdb command you must execute? You may assume that the name of file `fact.cpp` **(1 point)**

c. What does the following command do? `(gdb) start 5` **(1 point)**

4. You have implemented a recursive matrix multiplication that works perfectly when the matrix size is in power-of-two (shown below). You wish to store the input matrices in Z-order to get excellent locality. Here, A_{ij} and B_{ij} are block matrices of size n/2 x n/2 $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$

For an example 4x4 matrix shown next how would you store the matrix elements in memory? **(2 points)**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

5. Consider Gaxpy (y=y+Ax) and outer-product ($A=A+yx^T$) pseudocode shown below. **(6 points)**

gaxpy (a_j is j^{th} column of matrix A)

```
for j=1 to 16
 for i=1 to 16
   y[i]=y[i]+a_j x[j]
```

outer-product

```
for j=1 to 16
 for i=1 to 16
   a_ij= a_ij +y[i]*x[j]
```

a. Name the storage layout that is better for storing matrix A ϵ $\mathbb{R}_{16x16}$ above **(1 point)**

b. You wish to vectorize pseudocodes using AVX512 instructions (note: in AVX512 mode, a vector register can hold 16 `float` numbers). *For each of the pseudocodes shown*, list the number of required: i) vector load operations, ii) vector store operations iii) Assume that there exists a fused multiply-add unit and a special arithmetic instruction is available that can do element-wise multiplication of vectors and add the result to a vector. List the number of such special arithmetic instructions required. **(3 points)**

c. For A ϵ $\mathbb{R}_{nxn}$ $, x \in \mathbb{R}_{nx1}, y \in \mathbb{R}_{nx1}, and\ n\%16 = 0$ what is the computational intensity in each pseudocode? No credits for just writing the final answer. You must indicate the computational complexity and the data movement overheads involved. **(2 Points)**

Part II – take home: (6 points) Visit the discussion forum to receive the question paper and instructions.