Parallel Architectures and Programming Models, 2023W

Assignment 1: OpenMP Tasking, Roofline Model

GPSA :: Roofline Model

Include the following in your report of assignment 1:

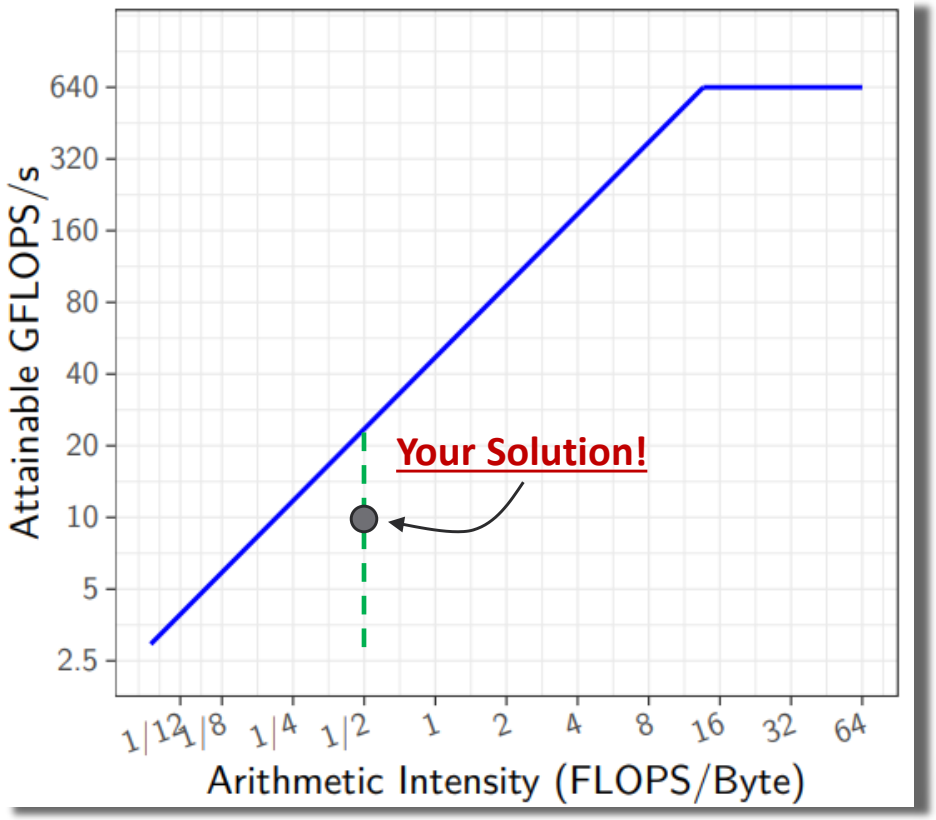
1) a log-log plot of the Roofline Model of a single computational node of ALMA;

2) include your best performing (32 cores) parallel implementation of GPSA to the model obtained in 1);

Based on the Roofline Model obtained above, and the Arithmetic Intensity (AI) of your solution, address the following questions in your report and justify your answers:

1) is your solution memory- or compute-bounded?

2) does it make sense to try to optimize your parallel solution? If yes, which optimization techniques could you employ?



About ALMA specifications...

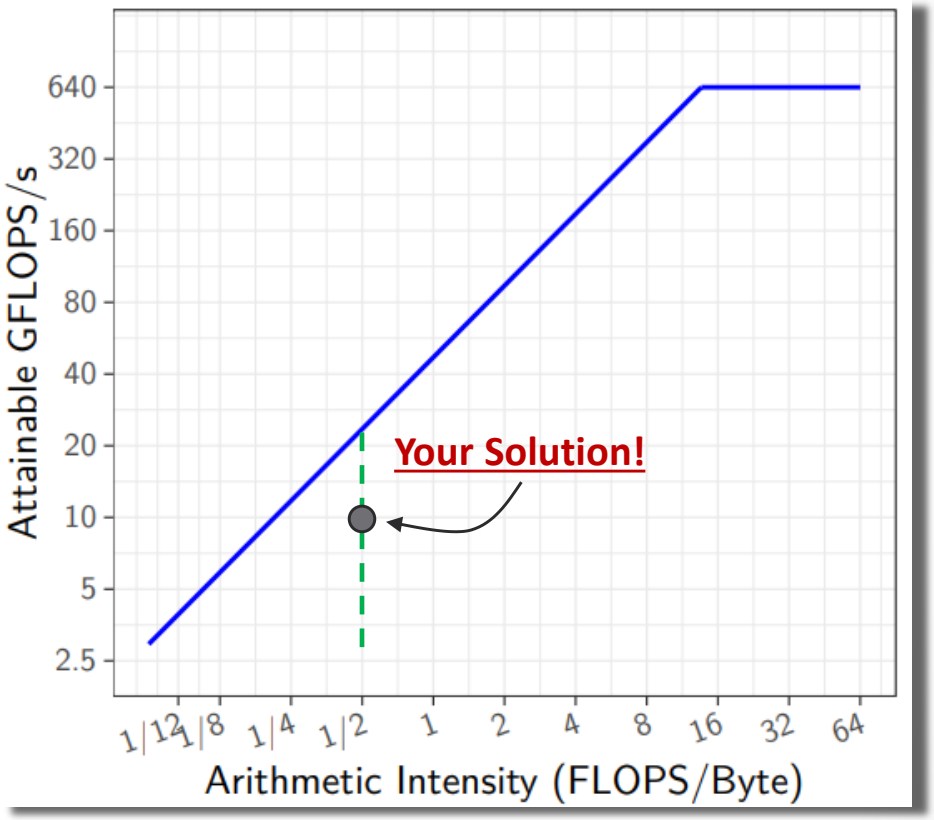
A single computational node of ALMA is composed by 2 Intel Xeon E5-2650 CPUs (with 2.0 GHz Clock Rate).

Each CPU has 8 physical cores, and each core can execute up to 2 AVX instructions per cycle:

- _mm_256_add_ps: 8 single-precision FLOPs (ADD)
- _mm_256_mul_ps: 8 single-precision FLOPs (MUL)

Additionally, the STREAM benchmark shows that the sustained memory bandwidth of each computational node is equal to 57 GB/s.

Note that the size of a single-precision floating-point number (`float`) is equal to 4 bytes.



About GPSA Arithmetic Intensity...

To compute the Arithmetic Intensity (AI) of your parallel solution (32 cores), you should only consider the code inside the innermost-loop of GPAS:

- for simplicity you should ignore initialization loops.
- If you modified the innermost-loop, calculate the Arithmetic Intensity of the code on the left instead.

You should also only consider floating-point operations (FLOPs), and ignore all int and char operations such as:

- e.g., `X[i-1]` reads a char and `cmap.at` reads an int.

For simplicity, you should ignore Read and Write operations to the substitution matrix SUB.

Lastly, assume that `std::max({match, del, insert})` requires 2 FLOPs per loop iteration.

```
for (unsigned int i = 1; i < rows; i++) {
    for (unsigned int j = 1; j < cols; j++) {
        //innermost-loop: BEGIN
        float match = S[i-1][j-1]
        + SUB[cmap.at(X[i-1])][cmap.at(Y[j-1])];
        float del = S[i - 1][j] + gap_penalty;
        float insert = S[i][j - 1] + gap_penalty;
        S[i][j] = std::max({match, del, insert});

        visited++;
        //innermost-loop: END
    }
}
```

Additional Information

- **The Roofline part of the assignment accounts for 20% of Assignment 1 grade.**
- **All information required to plot the Roofline model is either provided:**
  - in this document **("About ALMA specifications"),**
  - or can be obtained by running your parallel solution **(#FLOPs executed, execution time, …).**

- **You should NOT use profilers to obtain information (PAPI, PERF, LIKWID,…)!**
- **Include a table in your report with the values you used to plot the Roofline Model:**
  - **E.g., #FLOPs executed, execution time, PeakFLOPs/s, etc…**

- **The deadline for this part of the assignment is the same as the first part! (23 of November)**