ROOFLINE PAPER REPORT

~Shreyas S (EP22B040)

Question 1. Provide a short (100-word) summary of the paper.

The paper proposes a comprehensive performance model that offers insights on parallel software and hardware. It compares the performance as throughput against the operational intensity over a log-log plot to obtain the bounds on peak achievable performance. Acknowledging that peak-performance is unrealistic in many cases, the model allows visualizing 'performance-ceiling' inferring the impact of each optimization (TLP/ILP/SIMD/prefetching etc), to the extent of obtaining incremental benefits due to each optimization performed. With a case-study of application of the model to 4 kernels from the '7-dwarfs' to four multicore designs, the Ridge point on the roofline curve proved to be a better indicator of performance than clock rate or peak performance.

Question 2. What did you learn from the paper?

- The general principle of *bound and bottleneck* analysis
    - Instead of analysing a particular performance metric or parameter directly, the system bottleneck can be understood and quantified.
- The roof-line model itself gives a great visualization of the parallel-processor. It captures both the memory-bandwidth and core's performance constraints into one plot.
    - It also allows naive comparison of 2 processors with same DRAM bandwidth and can predict when exactly would there be a performance gain (based on the workload). This is possible because the fixed bandwidth ensures the same 45° line for peak memory bandwidth.
- For superscalar architectures, the highest performance comes when fetching, executing and committing the maximum number of instructions per clock cycle, which can be improved by increasing Instruction Level Parallelism.
- Usually, the cores have an equal number of adders and multipliers. Hence simultaneous floating-point additions and multiplications typically achieve the peak floating-point performance.
- Optimizing for unit stride, ensuring memory affinity and using software prefetching (in the order, unless computations of sparse data) ensures incrementally better memory-constraint performance.
    - Such optimizations allow for dividing the roofline plane into several segments. Based on the kernel, it is then possible to predict the exact segment in which the performance can be achieved.

- In general, the operational intensity of the kernel is to be increased before trying out any other optimizations, especially when lying in the memory constrained region of the roofline.
- In the context of AI workloads, most kernels would exhibit sparse-MV operations, whose operational intensity lies below the ridge point of all 4 multicores. Hence, most of the optimizations should involve the memory subsystem.
- A processor with large-bandwidth and easy to understand cores is the easiest to program, with low ridge point. AI kernels in such processors can just obtain good performing cores from the compiler and then use as many threads as possible.
    - The computer with the highest ridge point would typically have the lowest unoptimized performance, inspite of having a much higher peak compute performance.
- The Ridge point on the roofline curve proves to be a better indicator of performance than clock rate or peak performance.

Question 3. What did you like about the paper?

- The false hope or misleading of certain performance metrics like frequency or peak throughput can be mitigated through the roof-line plot of the processor.
- While our focus is restricted to Sp-MV operations for the most part, the case study also indicates how well the accelerator designs can perform for different kinds of workloads (for non-AI applications).
- Multiple processors with the same bandwidth, would have identical roofline for the peak memory constrained part, since its just a 45° line.

Question 4. What did you dislike about the paper?

- The incremental performance of optimizations can be truly appreciated only if the ease of performing them is known before-hand.
    - Hence, it is not possible to compare different orders of optimizations, segment-wise for different kernels in the same plot.

Question 5. Given a computer, how would you determine its peak memory bandwidth and its peak floating-point performance?

- The peak memory bandwidth can be obtained in 2 ways:
    - Either from reading through the datasheets provided and making the required calculations based on bus-width and the DRAM architecture itself.
    - Or by calculating it by finding the bus width, memory speed and data rate from the performance tab (in task manager, in the case of Windows OS).
- The peak floating-point performance can be estimated using the performance counters using open-source tools like *perf* that would report the execution time

taken for the number of floating-point operations performed. This can be used to estimate the throughput (Flops / second)

Question 6. In the slanted line of the roofline model (for the peak memory bandwidth), what does the y-intercept represent?

- Considering that it is a log-log plot, the y-intercept shown in the plots is not the true y-intercept. Ie: it doesn't correspond to the value of y when the corresponding x-axis value = 0.
- Hence, the coordinate axes are drawn arbitrarily for some value of operational intensity as x, usually <<1.
- Therefore, the y-intercept as seen in the curve just represents the maximum achievable performance for the corresponding x-axis value.
- However, if the y-intercept is taken in the ***log(performance throughput) vs log(operational intensity)***, the x=0 represents operational intensity of 1 Flop/byte. Therefore, the performance throughput in Flops/s denotes the **Memory Bandwidth** itself since (operational intensity * memory bandwidth) gives the slanting line of the roofline model.