Work assignment in AA

Context

The main goal of this assignment is to develop in students transversal skills applied to a specific topic in the Curricular Unit *Advanced Architectures*: the methodology on the characterization of the performance bottlenecks on each computing platform and on the code profiling and its performance analysis on that platform.
The development of these skills will be achieved through training in literature search, reading & interpreting scientific papers, planning experimental work, synthesizing relevant information, writing a short essay on a given theme and a short (10 min) oral communication and discussion of the results (on the 1st December).

You should start by reading a paper that proposes a very popular performance model, known as Roofline, the Amdahl law applied to heterogeneous multicore platforms, and the respective Gustafson extension to this law. Once read and understood their contents, you will characterize your own laptop based on the supplied information on the papers.
The next step is to get acquainted with one of the most popular portable interfaces to hardware performance counters on current processor architectures (in the form of a library), the *Performance API* (PAPI), by reading the paper that addresses this approach.

Once you carefully read these papers related to performance analysis and measurement of programs and computer systems, you are required to perform some specific tasks and to prepare a short presentation interpreting the obtained results.
This work must be performed by a 2-student team, who will deliver a single report and a single oral presentation.

Task 1
Full characterization of the hardware platform

1.1 Fully characterize your team laptop: manufacturer, model, CPU chip manufacturer/model/reference, main memory latency and size; for the CPU give more details on #cores, peak FP performance; for the memory hierarchy, present cache details and the memory access bandwidth (from the LLC on chip). Explain how you got these figures.

1.2 Build and compare the roofline model for single-precision FP on two computer systems: your team laptop and a dual Xeon cluster node (any 431 node). Show in the same operational-intensity graph the roofline for both systems; follow the suggestions in Appendix A of the 2008 paper on Roofline.

1.3 Add ceilings to the reference roofline model of your team laptop, as suggested in the paper, and clearly justify each ceiling. Order these ceilings according to a given kernel: the matrix multiplication.

Task 2
Operational intensity study of matrix multiplication algorithms

2.1 Install PAPI on your team laptop and identify all performance counters that are available for the system CPU. From these, select the most relevant ones to analyse an application execution time and identify potential bottlenecks. Note that PAPI is not

available yet for Mac OS; if both team members have only MacBook, you may install Ubuntu or use a 431 cluster node.

2.2 Write a single-threaded C function that computes the dot product of 2 square matrices with size NxN, **C = A * B**, in single precision, and with no block optimization. The function receives as arguments the pointers for the 3 matrices and their dimension N. The algorithm for this product contains 3 nested loops for the indexes i, j and k.

2.3 Analyse a different implementation of this triple nested loop, exploring one of the 6 alternative combinations of the index order: **(1)** i-j-k, **(2)** i-k-j, **(3)** j-i-k, **(4)** j-k-i-, **(5)** k-i-j, **(6)** k-j-i. The allocation of this index order will be allocated by an email sent to each team.
For each alternative implementation, access to the elements of either A or B (or both) will be row by row, or column by column, which may impact performance; to analyse and eventually reduce this negative impact, compare the original version with another one, where you transpose at the beginning the matrix(ces) that is(are) accessed by column, so that the reading accesses are performed row by row during the dot product computation. If your original version always accesses the matrices row by row, compare with a different index order where you have to transpose, at most, one matrix.

2.3.1 Select the following sizes for your data structure(s): 1 that will completely fit in L1 cache, 1 only in L2 cache, 1 only in L3 cache (if available) and 1 only in RAM.
2.3.2. Validate your code building a square matrix A with randomly generated values and a matrix B where all elements are "1"; in the product A*B all resulting columns have the same values, while in the product B*A all resulting rows have the same values.
2.3.3 Execute the code following the *K*-best scheme, with *K*=3 with 5% tolerance and at most 8 times; select the best execution time.
2.3.4 For the best execution times, and using PAPI data from the hardware counters,
2.3.4.1 Estimate the number of RAM accesses per instruction and the number of bytes transferred to/from the RAM; confirm those values with PAPI readings;
2.3.4.2 For each data set: (i) estimate the number of FP operations executed and (ii) plot the achieved performance of your code in the roofline performance graph;
2.3.4.3 Build a table where each line shows the %miss rate on memory reads in cache levels 1 and 2 (and 3 if available), for each data set size.

2.4 This function contains the right ingredients for vectorization. Compile your code with the adequate compiler switch and confirm it vectorized the code; if not, modify the code to force the compiler to vectorize. Repeat 2.3.3 and 2.3.4 only for the smaller data set.

2.5 Interpret the obtained results for the algorithms, starting with bound characterization (CPU bound or memory bandwidth bound), performance bottlenecks and the impact of the matrix transpose approach to structure data in memory.

Task 3
Report writing and oral presentation

Write a short essay (no longer than 6 pages plus annexes with additional info) describing the experimental setup and relevant results with associated discussion. Include a Title, list of authors, an Abstract, an Introduction, relevant mid-sections, a Conclusion and References (by order of appearance in the essay, with all data pertinent to find the publication, e.g. author(s), title, place where it was published and who published, year).
Deadlines for the essay: title & abstract **30-Nov-15**, full essay **08-Dec-15**.
The oral presentation, 10 min long, is scheduled for **01-Dec-15, 09h00 - 12h00**.