MEI/PDC

Work assignment in CSP *(19-Nov-12)*

Once you carefully read the supplied papers related to performance analysis and measurement of programs and computer systems (the 07-Nov-12 Reading assignment, see below), you are required now to perform some specific short tasks and to prepare a short presentation interpreting the obtained results.
This work must be performed by a 2-student team, which will deliver a single report and a single presentation.

Task 1
Full characterization of the hardware platform

1.1 Fully characterize your own 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 bandwidth of memory access from the higher cache level on chip.

1.2 Build the roofline model for both laptops in your group and show in the same operational-intensity graph either (i) your laptop models if they are different, or (ii) our team laptop compared with the Opteron X4 (4 cores); follow the suggestions in Appendix A of the paper on Roofline.
Add more ceilings to your laptop roofline model as suggested in the paper, and clearly justify each added ceiling.

Task 2
Operational intensity study of matrix multiplication algorithms

2.1 Install PAPI on your team laptop and identify all performance counters that you have available on the laptop. From these, select the most relevant ones to analyze 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, an alternative solution will be provided.

2.2 Consider a C function that computes the product of 2 square matrices with size N x N, 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 Each group will analyze a different implementation of this triple nested loop, exploring 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. 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 analyze 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.

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 at least 3 times for each data structure size (and at most 8 times), and select the best execution time that must be in a range no larger than 5% of the other 2 best values.

2.3.4 For the best execution times, and using PAPI data from the hardware counters,

2.3.4.1 Estimate the number of memory accesses per instruction and confirm that value with PAPI readings

2.3.4.2 Plot the following data: #instructions / byte_accessed_to_RAM; %miss rate on memory reads in cache levels 1 and 2

2.4 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 4 pages) describing the experimental setup and relevant results with associated discussion.
The oral presentation is scheduled for **27-Nov-12, 11h00 - 13h00**.

Reading assignment in CSP *(07-Nov-12)*

Read_me_first

The main goal of this reading assignment is to develop in students transversal skills applied to a specific topic in the module *Computer Systems and Performance*: 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 27th November).

The starting material for this reading is a paper on the performance model known as Roofline, complemented with the Gustafson extension and with a revision of the Amdahl law applied to heterogeneous multicore computers. Everyone should read it and try to characterize its own laptop based on the supplied information on the paper.
The next step is related to get acquainted with one of the most popular portable interface (in the form of a library) to hardware performance counters on current processor architectures, PAPI, by reading one of the papers that address this approach.

Later on we will supply the specification of the group/individual activities, through a revised edition of this document.