CPE 631 Advanced Computer Systems Architecture: Homework #2

The goal of this assignment is to expose students to various performance evaluation tools and approaches, specifically time measurement and PAPI.

Problem #1 (40 points) Measuring time, Compilers

Write a C program that multiplies two squared matrices A and B of size N (NxN). The matrices consist of double-precision elements. The program should accept the matrix size, N, as a parameter; initialize the matrices using a random number generator; and find the resulting matrix C=A*B. The resulting matrix should be written to a binary file (please delete this file after the execution of your program).

Create two executables, one created by the gcc compiler (optimization level –3) and one created by the Intel compiler (with optimization level –fast).

(a) Measure the entire program execution times as a function of matrix size (N=512, 1024, 1536, and 2048). Plot your results as a function of N (plot should include two lines, one for the gcc executable and one for the icc executable).

(b) Instrument your code and measure the execution time *of the critical loop that finds the resulting matrix*. Plot your results as a function of N.

Submission should include the following:

(i) Source code for the base matrix multiplication from (a) and (b); Note: make your code as simple as possible and do not submit/reuse the code that is given to you in CPE 512;

(ii) A graph showing the execution times for both executables as a function of N from (a);

(iii) A graph showing the execution times of the critical loop for both both executables from (b);

(iv) Analysis of the results (do the results meet your expectations, what are trends in the plots, etc).

Readings:

(i) Time measurement tutorial can be found in: http://lacasa.uah.edu/portal/index.php/tutorials/30-measuring-program-execution-time;

(ii) Using Intel compiler tutorial can be found in Canvas: Files/Tools/NotesonUsingIntelCompiler.txt.

Problem #2 (30 points). Parallel matrix multiplication

Design a new matrix multiplication program that uses OpenMP to parallelize the critical loop. Create two executables, one with the gcc compiler and one with the icc compiler. Compare the execution times of the critical loop before parallelization (from Problem #1) and with parallelization. Plot your results as a function of N (your plot should include four lines: Serial.GCC, Serial.ICC, OpenMP.GCC, OpenMP.ICC.

(i) Source code for the matrix multiplication with OpenMP;

(ii) A graph showing the execution times of the critical loop for four executables;

(iii) Analysis of the results (do the results meet your expectations, what are trends in the plots, etc). Please note that your findings would depend on the host machine you are running your program on.

Problem #3 (40 points) PAPI

Instrument your codes from Problem #1 and Problem #2 with PAPI calls and measure the following events for the critical loop: (i) the number of instructions executed (PAPI_TOT_INS), (ii) the total number of clock cycles (PAPI_TOT_CYC), (iii) level 1 data cache misses (PAPI_L1_DCM), and (iv) the total number of L2 misses (PAPI_L2_TCM) in the critical loop. For each event, plot a separate graph with lines for each executable. Analyze the results.

Submission should include the following:

(i) PAPI instrumented code for serial matrix multiplication;

(ii) PAPI instrumented code for the parallel matrix multiplication;

(iii) 4 graphs showing the PAPI events as a function of N and the type of executable;

(iv) Analysis of the results.

Readings:

(i) http://lacasa.uah.edu/portal/Upload/tutorials/papi/PAPI.pdf