Programming for performance on multi-core and many-core

LS2

Performance of Scalar Add

This problem builds on the study of ILP using reductions in the Lecture. Download, extract and inspect the code. The C source and header files in the scalaradd folder benchmark this kernel under Linux (x86 compatibles). (a) Compile and run the code. the code outputs a file containing different results for varying number of accumulator variables (K) and unrolling factors (L). Summarize the results in a table similar to the one shown in Lecture. For which K do you obtain the maximal performance? Compare against the model K = d latency cycles per issue e.

(b) The file funcs.h contains different functions for every possible pair of K and L. Pick the one associated to the best values of K and L for the exercise explained now.

i. For all vectors of sizes n = 100 . ceil($1:8^i$), with i between 1 and 21, create a semi-log plot where the x-axis is in logarithmic scale, and the y-axis in normal scale. The x-axis shows n and the y-axis performance in Mflop/s or Gflop/s.

ii. You should see that the line consists of piecewise plateaus. Try to explain the plateaus, meaning try to explain the height of the plateaus and where the plateaus transition using suitable hardware parameters.

LS3: Brach less merge sort

Branchless Mergesort:

Write code for sorting huge data (hint: use long ints) using the merge sort algorithm. Can you optimize the code such that the branches are optimized away. Write a report why the performance is impacted?

Use the following optimizations and report the timings, branches, branch-misses, instructions, cycles etc. Check if the conditional move instruction is being used?

[Use the code http://rosettacode.org/wiki/Sorting_algorithms/Merge_sort#C]