Optimizing the Fast Fourier Transform on a Multi-core Architecture

Critique by
Saleel Kudchadker

This paper presents a study on the challenges based on the optimization of the Fast Fourier Transform (FFT) on the IBM Cyclops-64 chip architecture. The paper has successfully implemented 1D and 2D FFT computations to optimize problem decomposition, load balancing, work distribution, and data-reuse, together with the exploiting of the C64 architecture features such as the multi-level of memory hierarchy and large register files.

Strengths:

- The paper has clearly indication how the FFT algorithm has been optimized step by step.
- The FFT computations are optimized by reducing the number of unnecessary operations.
- The paper has highlighted various ways of tweaking the processor to increase the performance like register renaming, Loop Unrolling etc.
- The number of computations for 2D FFT is N^2logN which are reduced by at least a factor of 2 for normal calculation.

Weaknesses:

- Among the questions raised in class were the scratchpad memory is not being used for optimization.
- All the CPU registers get used during computation of 2 point DFT, which increases the memory cycles. The paper also doesn’t explain methods, when the FFT size is increased so that the input values exceed the number of registers.
- The processes of optimization have not been fully automated and manual application is required especially in register renaming and instruction scheduling when optimizing the kernel functions.

Overall the paper does in-depth analysis of optimizing FFT calculation on Cyclops-64. It shows that the application development on multicore architecture is relatively difficult. Improving performance is problem specific and needed various tweaks to achieve good results.